



Final Report for the

CWR Global Portal

[<http://cwrint.grinfo.net>]

Prepared by:

Dag Terje Filip Endresen

Nordic Genetic Resource Center (NORDGEN) / Bioversity International

Email: dag.endresen@nordgen.org, d.endresen@cgiar.org

CWR Portal

Home About CWR Data Feedback Help Site map Search!

About

- Project Partners
- National Inventories
- External Datasets
- Data sharing and use agreements
- Portal technical specifications
- CWR Links
- Search the CWR Portal web site

CWR Data

Welcome to the Crop Wild Relative Global Portal

This portal provides access to information and data resources important for the conservation and utilization of crop wild relatives (CWR). It was created within the UNEP-GEF supported project *in situ* conservation of crop wild relatives through enhanced information management and field application. The development of the portal is ongoing and new resources are being added as they become available.

Latest Images

A selection of the latest images added to the [CWR Image Archive](#). Please [contact us](#) if you want to contribute images of Crop Wild Relatives.

Latest News Stories

- CWR Portal (2007-01-01)**
The CWR Portal is developed by Bioversity International as a generic web application written in PHP and with a generic ADODB database connection to the PostgreSQL database system. Some of the routine operations have also been coded for the Perl scripting language. The portal web application have been successfully tested with the Apache web server for the Apple Mac, Linux and Windows operating system environment.
- CWR Project (2004-01-01)**
The crop wild relatives global portal has been established within the framework of the UNEP/GEF supported project *In-situ* conservation of crop wild relatives through enhanced information management and field application, lead by Bioversity International and implemented from 2004 to 2009. Funds for the development of the global portal were also received from the German Federal Ministry for Economic Cooperation and Development (BMZ).

[View all News items](#). A total of 2 news stories have been registered.

Page last modified : 2007-11-07 23:27:52 +0100 (Wed, 07 Nov 2007) Current time is: 2007-11-09 13:42:07

[\[Contact\]](#) [\[Feedback\]](#) [\[CWR Portal terms of use\]](#) [\[Terms of use external datasets\]](#) [\[Citation\]](#) You are not logged in! [\[Login here\]](#)

IPGRI and INIBAP operate under the name Bioversity International.
 © Bioversity International - Headquarters: Via dei Tre Denari, 472a 00057 Maccarese (Rome) Italy
 Tel.: (39) 066118.1 - Fax: (39) 0661979661

Figure 1, screenshot of the CWR, CropWild Relative Global Portal, [<http://cwrint.grinfo.net>]

The CWR global portal provides access to information and data resources important for the conservation and utilization of crop wild relatives (CWR). It was created within the UNEP-GEF supported project *In situ* conservation of crop wild relatives through enhanced information management and field application. The development of the portal is ongoing and new resources are being added as they become available.

The germplasm data portal is a generic data portal application for integration and online publication distributed datasets based on the SESTO genebank information system developed by the Nordic Genetic Resource Center (NORDGEN) previously called Nordic Gene Bank (NGB). The generic data portal application is also based on the GCP Central Repository and the Germplasm Clearing House Mechanism (CHM) both developed by Bioversity International.



Germplasm Data Portal

A generic data portal application for distributed datasets

Technical description of the germplasm data portal application
Dag Terje Filip Endresen
Document last updated: February 19, 2008

Index

Index	4
List of figures.....	5
Introduction.....	8
Source code	8
Source code, directory and file structure	8
INFO.TXT	9
Example: how is the display of the welcome home page implemented?	12
Object primary keys as URL GET attributes.....	13
Getting started with a new data portal implementation.	13
The layout elements	13
HTML HEAD	14
CSS, Cascading Style Sheet.....	15
Page menus	15
Page content frame.....	16
Information pages	17
Sub applications	17
Data harvest routines and methods	18
Datasets provided as a XML web service (BioCASE)	19
Dataset(s) provided as a REST XML web service (GBIF).....	23
The GBIF data portal offer REST web service interfaces for taxon, occurrence records, occurrence density, dataset metadata, data provider metadata and data network metadata level data. An example of the occurrence record REST service request style:	24
An example of the service request style asking for all occurrence records of the species <i>Allium porrum</i> :.....	24
An example of the service request style asking for all occurrence records of the species <i>Allium porrum</i> with geospatial origin attributes reported (geo-referenced records only):	25
Datasets provided as a simple file.....	29
Import of external datasets.....	30
Step 1, download dataset file from online source URL.	32
Step 2, un-compress the source dataset file, if needed (zip, tar, gz, bz2).	32
Step 3, Convert the (un-compressed) dataset file to tab-separated text.....	32
Step 4, recode the tab-text dataset file to Unicode, if needed.....	32
Step 5, transform the tab-delimited dataset file to SQL INSERT script.....	33
Step 6, IMPORT dataset to the database	33
Import of external reference datasets (examples from the CWR Global Portal).....	34
WIEWS Institute.....	34
BGCI Garden and BGCI Plants	34
IUCN Red List	35
WDPA, World Database on Protected Areas	35
EURISCO	35
SINGER	36

Taxon and country unit level summary metadata	36
Manual update of taxon and country unit level metadata	39
Data dictionary	41
Frequently asked questions:	43
Software used by or useful to the data portal.....	43
References:.....	44

List of figures

Figure 1, screenshot of the CWR, CropWild Relative Global Portal, [http://cwrinfo.net]	2
Figure 2, file directory showing the data portal root directory.	8
Figure 3, chain or sequence of scripts to display the welcome home page.	12
Figure 4, file directory showing the content of the “data_portal/page_elements/” folder..	14
Figure 5, file directory showing the content of the “data_portal/page_elements/cwr/” folder.	14
Figure 6, page application menu, level 1 (page_menu_1.phps).	15
Figure 7, left side menu (page_menu_left.phps).	16
Figure 8, the page middle content frame wraps the data portal content from a sub- application or from a information web page.....	16
Figure 9, information web pages are loaded from the “data_portal/webpages/<scope>” directory, requested by the \$_REQUEST[‘page’] GET attribute.	17
Figure 10, portal content sub applications are loaded from the “data_portal/applications/” directory, requested by the \$_REQUEST[‘app’] GET attribute.....	18
Figure 11, the first version of the data portal was the Germplasm Clearing House Mechanism, designed to access, scan and index XML data from BioCASE database wrapper web service end points.....	19
Figure 12, step 1 of the CHM is a list of data provider BioCASE service end points. All the BioCASE DSA URLs are registered to provide the starting point for a data harvest session. A normal UDDI with a standard WSDL style discovery would be a useful extension of this step 1.....	20
Figure 13, step 2 is the list of supported global data standards including their mapping to the implemented CHM data model of the CHM database index.	20
Figure 14, step 3 is the interface to formulate the data request (request.xml) according to the BioCASE protocol. The data harvest methods are developed as a PHP library and can be started either directly from the web interface or from the UNIX prompt command line (or the crontab). The data harvest includes paging of the XML data response from the harvested BioCASE end point when there are more records available than the requested number of records per page (or the maximum allowed records per page the remote BioCASE DSA is configured to allow).....	21
Figure 15, step 4 is the preview of the harvested XML data, extracting selected data values and the import of these values to the CHM database index.	22
Figure 16, the CHM portal also comes with a search interface to the CHM database index.....	23

Figure 17, the Global Biodiversity Information Facility (GBIF) maintains a data portal of global distributed datasets on biodiversity based on the standards developed and maintained by TDWG (Biodiversity Information Standards).	24
Figure 18, example of GBIF response format: [http://data.gbif.org/ws/rest/occurrence/count?scientificname=Allium+porrum]	25
Figure 19, example of GBIF response format: [http://data.gbif.org/ws/rest/occurrence/count?scientificname=Allium+porrum&georeferencedonly=true&stylesheet=]	26
Figure 20, this is the PHP code to access the GBIF data portal REST web service interface.....	27
Figure 21, the function in the previous figure (Figure 19) to refresh the cached summary number of species occurrences from the GBIF web service can be invoked from the germplasm data portal web interface.	28
Figure 22, the function (Figure 19) to refresh the count of occurrence records for a species from the GBIF REST web service can be invoked from a PHP script “data_portal/applications/import_datasets/import_gbif_taxon.phps”. This script can be executed from the command line or added to the crontab for a scheduled automatic refresh (... may require some minor update of the current version of the script).	29
Figure 23, the configuration attributes for the “import dataset” sub-applications, showing the attributes for the WIEWS Institute as example.....	31
Figure 24, a summary flow of the steps to access, download, convert and import an external reference dataset to the germplasm data portal.	33
Figure 25, search interface (simple keyword search) for taxon level metadata from the indexed external datasets on CWR resources.	36
Figure 26, example of a taxon level metadata detail page for <i>Allium schoenoprasum</i>	37
Figure 27, search interface (advanced search) for country level metadata from the indexed external datasets on CWR resources.	38
Figure 28, example of a country level metadata detail page for Italy.....	39
Figure 29, example of using the PostgreSQL database prompt to update table data.....	39
Figure 30, here is the link to the “edit country metadata” form. This link is ONLY displayed for logged in users.	40
Figure 31, this is the edit form for country level metadata. You would normally update these data points from the (semi-) automatic update routines for external datasets. For example the GBIF summary metadata is very easy to update (per unit as well as for more units) from the link located directly next to the link to this form from the taxon and country level metadata detail pages... ..	40
Figure 32, example of descriptive column names and mouse over column tip as defined from the data dictionary for a data unit list view.	41
Figure 33, example of descriptive column names from the data dictionary for a data unit detail view.	41
Figure 34, start the data dictionary description by a description of the database table (step 1).	42
Figure 35, next describe the individual columns using the [Edit DM] links from the table description detail page. You may also consider updating the column description	

from the [Edit] link as well. Work is in progress for a new improved data dictionary
model based on this concept. 42

Introduction

The germplasm data portal is developed by Bioversity International and the Nordic Genetic Resource Center (NORDGEN) as a generic web application written in PHP (version 5) and with a generic ADODB database connection to the PostgreSQL database system (PostgreSQL version 8). Some of the routines and methods have also been coded using the Perl scripting language. The portal web application has been successfully tested with the Apache web server (version 2) for the Apple Mac OSX, Linux and Windows XP operating system environment.

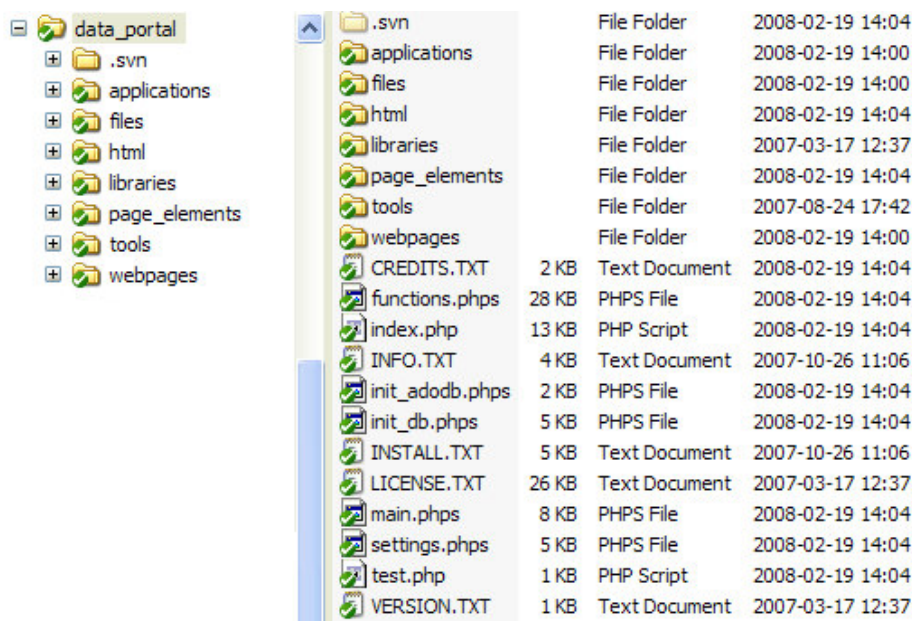
The portal web application is based on the SESTO genebank system developed by NORDGEN. The CWR Global Portal is actually only one layout skin of the very same portal application as used for the Svalbard Global Seed Vault data portal, the Generation Challenge Program Central Registry, and the ECPGR ECCDB databases hosted from NORDGEN ...with more.

Source code

The [data portal source code](http://wwwdev.ngb.se/WebSVN/listing.php?repname=data_portal) is available from the Subversion code repository hosted by NORDGEN.

[http://wwwdev.ngb.se/WebSVN/listing.php?repname=data_portal]

Source code, directory and file structure



The image shows a screenshot of a file directory. On the left, a tree view shows the 'data_portal' directory expanded, revealing subdirectories: .svn, applications, files, html, libraries, page_elements, tools, and webpages. On the right, a detailed list view shows the contents of the 'data_portal' directory. It includes folders like .svn, applications, files, html, libraries, page_elements, tools, webpages, and various text and PHP files with their sizes and modification dates.

.svn	File Folder	2008-02-19 14:04
applications	File Folder	2008-02-19 14:00
files	File Folder	2008-02-19 14:00
html	File Folder	2008-02-19 14:04
libraries	File Folder	2007-03-17 12:37
page_elements	File Folder	2008-02-19 14:04
tools	File Folder	2007-08-24 17:42
webpages	File Folder	2008-02-19 14:00
CREDITS.TXT	2 KB Text Document	2008-02-19 14:04
functions.phps	28 KB PHPS File	2008-02-19 14:04
index.php	13 KB PHP Script	2008-02-19 14:04
INFO.TXT	4 KB Text Document	2007-10-26 11:06
init_adodb.phps	2 KB PHPS File	2008-02-19 14:04
init_db.phps	5 KB PHPS File	2008-02-19 14:04
INSTALL.TXT	5 KB Text Document	2007-10-26 11:06
LICENSE.TXT	26 KB Text Document	2007-03-17 12:37
main.phps	8 KB PHPS File	2008-02-19 14:04
settings.phps	5 KB PHPS File	2008-02-19 14:04
test.php	1 KB PHP Script	2008-02-19 14:04
VERSION.TXT	1 KB Text Document	2007-03-17 12:37

Figure 2, file directory showing the data portal root directory.

The data portal source code is easily installed simply by extracting the source code directories and files to a folder on your local server (or a desktop/laptop workstation with the Apache web server and PHP5 installed). Apache httpd [<http://httpd.apache.org>], PHP [<http://www.php.net>]. You may extract the data portal source code to any folder you want as the application use a relative path to refer to internal scripts (see Figure 2). Second you will need to mount the “data_portal/html/” directory to the public web tree (www) of your web server. You may mount the html folder anywhere you want in your public web tree as the data portal use relative internal URLs to reference internal resources. Only the “html” directory should be mounted not the entire “data_portal” directory folder. If you mount the entire “data_portal” directory the data portal will still work as normal, but you will also publish (online) all the source code including your configuration files with usernames and passwords with more.

The data portal is divided in modules inspired by the “cascading style sheet type” logic. The execution of the portal PHP scripts starts at the higher directory level and follow the path down the directories to child directories for more specific features or functionality. For example general configuration settings are provided at the higher directory level (data_portal/settings.phps), more specific settings are included for the layout page elements (data_portal/page_elements/settings.phps), with even more specific layout settings for the CWR portal implementation in a sub-folder with the same name as the implementation scope (data_portal/page_elements/cwr/settings.php). The same pattern of a subfolder for specific implementations for a particular data portal implementation (like the CWR, SGSV, SESTO, EAPGREN, ECPGR ECCDB etc) is repeated through the portal source code. E.g. the CSS for the CWR is in a sub-folder named “cwr” (data_portal/html/css/cwr/style.css), the images of the CWR image archive is saved to a sub-folder named “cwr” (data_portal/html/image_archive/cwr/) etc.

INFO.TXT

All (or most of) the directories of the source code contains a file INFO.TXT with more detailed information about the files you find at each directory level. Below you see the INFO.TXT file of the data portal ROOT directory as an example.

Source code, ROOT directory

All content for the data portal is included from the “./html/index.php” page. The “./html/index.php” page does little itself than open the “./main.phps” script in the ROOT directory described here. The “./main.phps” script starts by reading the “settings.phps”, “init_adodb.phps” and “init_db.phps” scripts inside the ROOT directory folder.

- * settings.phps
- * init_adodb.phps
- * init_db.phps

The `./index.php` inside the ROOT directory is only a dummy page and should normally not be available from the web tree. When the data portal is properly installed only the `./html` folder will be mounted in the public web tree! Note that if the complete `data_portal` directory itself is unzipped or copied into the web tree, then all the settings and configuration may be publicly readable. This may be useful during development, but could be a serious security problem for a production version of the data portal!

Add the following lines to the `httpd.conf` of your Apache web server:

```
Alias /portal "path_to_the_data_portal_folder/html"
<Directory "/path_to_the_data_portal/html">
    Options Indexes
    AllowOverride None
    Order allow, deny
    Allow from all
</Directory>
```

The `"path_to_the_data_portal"` could for example be `"/usr/local/data_portal"`. The `httpd.conf` configuration could also be added to the extension folder, e.g. `"/usr/local/apache2/conf/extra/data_portal.conf"`. You will find an example file to copy here in the `./tools/httpd__conf.d/"` folder.

FILES in the ROOT directory

- * `./main.php`
The `./main.php` starts by reading the `settings.php`, `init_db.php` and `init_adodb.php` scripts inside the `./page_elements` folder. See `"INFO.TXT"` inside this folder for more information... Then `./main.php` will continue with building the HTML tags for `<html>`, `<head>` and `<body>`. Inside the body the content is collected from the sub folder `./page_element`.
- * `./settings.php`
The settings script in the ROOT directory is included in `./main.php` before the settings from the `page_element` sub-folder (`./page_elements/settings.php`). Here some generic system parameters are set or calculated from the web server environment variables.
- * `./init_adodb.php`
This script is used to include the ADODB database abstraction library.
- * `./init_db.php`
This script is used for configuration of the database connection for the ADODB database abstraction library included in the `"init_adodb.php"` script.

SUB-DIRECTORIES of the data portal ROOT

- * `./applications/`

The applications sub-folder is for the sub applications inside the page_content framed box. These sub-applications are included by the `_REQUEST['app']` parameter from the URL. The application with the directory name equal to the value of the `_REQUEST['app']` will be loaded. The data portal will start by looking for the script "main.phps" in this directory and give an error message if the directory is missing or if the directory does not have this script main.phps).

- * `./webpages/`
Simple information web pages are included from the `_REQUEST['page']` parameter from the URL. A web page with the extension ".inc" or ".phps" after the value of `_REQUEST['page']` will be displayed. If no such page, an error message will inform the user of this problem. The data portal will look for this page.inc or page.phps in the "data_portal/webpages/" directory.
- * `./files/`
The "files" folder in the ROOT directory is for data files. Here the data backup files for the database tables are saved. The data harvest and data import routines also save files here. The web server will need write permissions for this folder as well as to the relevant sub folders!
- * `./html/`
This folder is mounted in the web tree of the web server. Files, scripts and sub directories will be published (online) from your data portal implementation.
- * `./libraries/`
Code libraries are included here. This folder should be used much more!! Most of the functionalities of the sub applications in the "applications" folder should be rewritten as PHP classes and moved here! I hope to find the opportunity to do this, as the data portal will be significantly more robust with more functionality in PHP classes than in crude .phps scripts. Later versions of PHP (version 6+) may require that more functionality is done as object oriented classes...?
- * `./page_elements/`
All the layout elements are included from this sub-folder. The page banner, the page menus as well as some of the scope specific configuration files for database settings etc...
- * `./tools/`
This sub-folder includes some supporting tools and script. You will find a number of scripts coded in Perl or the bash shell scripting language (with more) in "tools/bin/*". You will also find some useful SQL scripts in the "tools/sql/*" folder.

READ MORE about the individual sub-folders from the INFO.TXT located inside each folder.

Example: how is the display of the welcome home page implemented?

Most users will first see the home page. This is the standard default if no further feature request is called. The public portal web root is mounted from “data_portal/html/” and the page index.php will be loaded (data_portal/html/index.php). This page (index.php) does nothing, but include the script “data_portal/page_elements/main.phps”. The “data_portal/page_elements/main.phps” creates the html page itself, loading the <html>, <head> and <body> tags. The page icon, the page top banner and menus are loaded from this “main.phps”.

```
./html/index.php
  ./main.phps
    ./settings.phps
    ./page_elements/settings.phps
    ./page_elements/cwr/settings.phps
    ./init_adodb.phps
      ./libraries/adodb/tohtml.inc.php
      ./libraries/adodb/toexport.inc.php
      ./libraries/adodb/adodb.inc.php
    ./init_db.phps
    ./page_elements/functions.phps
    ./page_elements/cwr/html_head.phps
    ./page_elements/cwr/page_menu_0.phps
    ./page_elements/cwr/page_banner.phps
    ./page_elements/cwr/page_menu_1.phps
    ./page_elements/cwr/page_menu_2.phps
    ./page_elements/page_content.phps
    ./webpages/cwr/welcome.inc
    ./page_elements/cwr/page_menu_left.phps
    ./page_elements/cwr/page_foot.phps
```

Figure 3, chain or sequence of scripts to display the welcome home page.

I have used the tab indent to indicate from which scripts the individual scripts are included. For example you will see that the data_portal/html/index.php itself only is responsible for include of the data_portal/main.phps, while this script (data_portal/main.phps) is responsible for including most of the other scripts. Only the data_portal/init_adodb.phps and the data_portal/page_elements/page_content.phps actually include script elements themselves. Most interesting is the page_content.phps as this script acts as a “content wrapper”. When the portal receives the GET \$_REQUEST[‘app’] or \$_REQUEST[‘page’] feature request the corresponding webpage or sub application will be loaded. Most of the sub applications will initiate an independent chain/sequence of included scripts from its own “main.phps” application wrapper. This way the sub applications can easily be moved to another content wrapper

(like for example sometimes even another format than the HTML web page environment).

Object primary keys as URL GET attributes

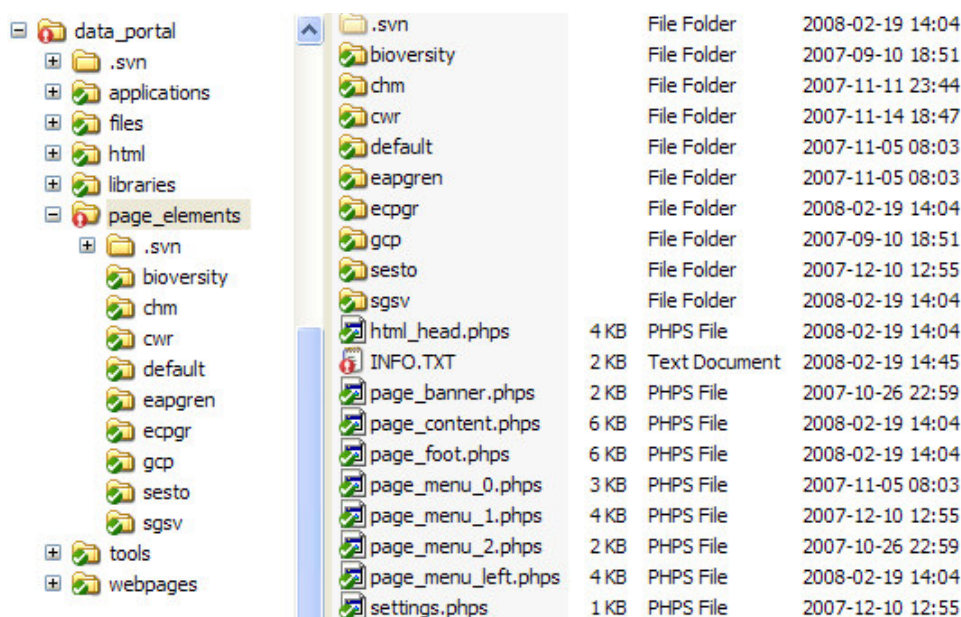
The portal will also respond when no “app” or “page” GET attribute is transmitted – if a data object key is transmitted as a URL GET attribute. For example
\$_REQUEST['taxon_id'], \$_REQUEST['taxon_name'], \$_REQUEST['country_id'],
\$_REQUEST['institute_id'], \$_REQUEST['person_id'], \$_REQUEST['image_id']
etc...

Getting started with a new data portal implementation.

To create a new data portal implementation you need to choose an acronym for your portal implementation. This acronym is named “scope” in the context of the portal scripts and used as the directory folder name for specific content and configuration for your new implementation. The concept of this scope is such that multiple portal implementations can “live” in these “scope-acronym” directory folders sharing the same base portal source code. The default scope acronym can for example be defined from the “data_portal/settings.php” configuration script in a similar manner as the presented examples for the CWR, SGSV, and EAPGREN implementations. This is only the default scope whereas the scope can thus be swapped by the user simply giving the URL GET attribute \$_REQUEST['scope']. (You may of course deactivate this behavior in for example the generic “data_portal/settings.php” configuration file if you do not wish to support user initiated scope swaps.)

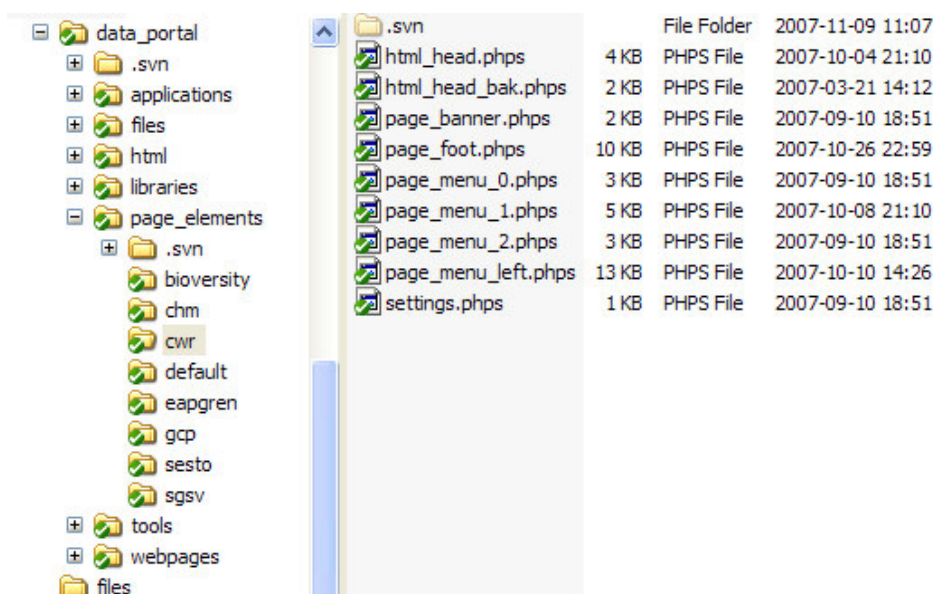
The layout elements

The layout elements are located in the ROOT sub-folder “page_elements” (“data_portal/page_elements/”) (see Figure 4). Here you will find the script defining the HTML page wrapper including the HTML META tags (“data_portal/page_elements/html_head.php”), the page banner (“data_portal/page_elements/page_banner.php”) and the page menus (data_portal/page_elements/page_menu_0.php, ...-menu_1.php, ...-menu_2.php, ...-menu_left.php) as well as the page footer (data_portal/page_elements/page_foot.php). Directly in this “page_elements” directory you will find the generic default example elements. Start by creating a subfolder in the “page_elements” directory with the same folder name as your chosen scope acronym, (see Figure 5). Next copy the default scripts you wish to modify from the “page_elements/” directory to your new scope directory. If the data portal application does not find the individual page element scripts in your scope folder, the default script will be loaded. Contrary if the scope specific script is successfully located the default page element script will not be loaded (there can only be one top menu, page banner etc...).



File/Folder	Size	Type	Modified
.svn		File Folder	2008-02-19 14:04
bioversity		File Folder	2007-09-10 18:51
chm		File Folder	2007-11-11 23:44
cwr		File Folder	2007-11-14 18:47
default		File Folder	2007-11-05 08:03
eapgren		File Folder	2007-11-05 08:03
ecpgr		File Folder	2008-02-19 14:04
gcp		File Folder	2007-09-10 18:51
sesto		File Folder	2007-12-10 12:55
sgsv		File Folder	2008-02-19 14:04
html_head.php	4 KB	PHPS File	2008-02-19 14:04
INFO.TXT	2 KB	Text Document	2008-02-19 14:45
page_banner.php	2 KB	PHPS File	2007-10-26 22:59
page_content.php	6 KB	PHPS File	2008-02-19 14:04
page_foot.php	6 KB	PHPS File	2008-02-19 14:04
page_menu_0.php	3 KB	PHPS File	2007-11-05 08:03
page_menu_1.php	4 KB	PHPS File	2007-12-10 12:55
page_menu_2.php	2 KB	PHPS File	2007-10-26 22:59
page_menu_left.php	4 KB	PHPS File	2008-02-19 14:04
settings.php	1 KB	PHPS File	2007-12-10 12:55

Figure 4, file directory showing the content of the “data_portal/page_elements/” folder.



File/Folder	Size	Type	Modified
.svn		File Folder	2007-11-09 11:07
html_head.php	4 KB	PHPS File	2007-10-04 21:10
html_head_bak.php	2 KB	PHPS File	2007-03-21 14:12
page_banner.php	2 KB	PHPS File	2007-09-10 18:51
page_foot.php	10 KB	PHPS File	2007-10-26 22:59
page_menu_0.php	3 KB	PHPS File	2007-09-10 18:51
page_menu_1.php	5 KB	PHPS File	2007-10-08 21:10
page_menu_2.php	3 KB	PHPS File	2007-09-10 18:51
page_menu_left.php	13 KB	PHPS File	2007-10-10 14:26
settings.php	1 KB	PHPS File	2007-09-10 18:51

Figure 5, file directory showing the content of the “data_portal/page_elements/cwr/” folder.

HTML HEAD

The HTML HEAD is defined from the “data_portal/page_elements/<scope>/html_head.php script. Here the HTML META tags are defined. The default page shows an example on how to define most of the relevant

Dublin Core definitions. The “html_head.php” is also where you define the link to the CSS style definitions you wish to use.

CSS, Cascading Style Sheet

As mentioned, the CSS style to be linked is defined in the HTML HEAD script. It is recommended that you keep the link to the generic “data_portal/html/css/style.css”. You may define your own CSS style from the “data_portal/html/css/<scope>/style.css” to override the generic style definitions. You may name your style sheet files as you wish as long as you provide the link to them from the “page_elements/<scope>/html_head.php” script. The portal application output is written as XHTML and with the aim of keeping all presentation layout definitions in the separate CSS file “style.css”.

Page menus

The top menu of the page is defined by the script “page_elements/<scope>/page_menu_0.php”. The two page application menus are defined by the “page_elements/<scope>/page_menu_1.php” and the “page_elements/<scope>/page_menu_2.php”. The menus are defined as a basic HTML bullet list. The CSS definition for ‘nav1’ and ‘nav2’ is used to transform the list to the horizontal menu as displayed, “menu item 1menu item 2...”. Add or remove list items to reflect the menu items you wish to have displayed. You may leave your scope version of the menu blank (no bullet list) to remove individual horizontal menus for your portal implementation. For example the CWR implementation have blank top menu “page_menu_0.php” and second level application menu “page_menu_2.php” (see Figure 6).

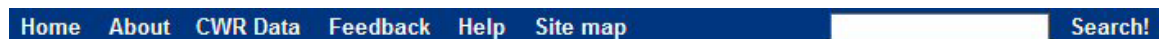


Figure 6, page application menu, level 1 (page_menu_1.php).

The left side menu items (see Figure 7) is defined in the same way by a bullet list and transformed by the CSS layout definitions for “portlet_left”, “portlet_title” and “portlet_content”.

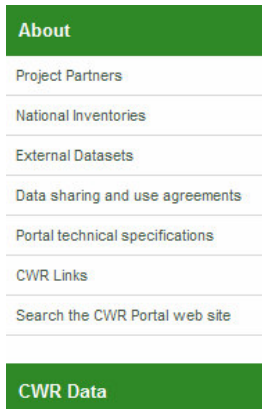


Figure 7, left side menu (page_menu_left.phps).

Page content frame

The page content frame itself is defined by the CSS definition for “content” of the <div id=’content’> element on the “./main.phps” script. Everything INSIDE this content div block frame is included from the “page_elements/page_content.phps” script. The content of the middle page frame can be either a page from the “data_portal/webpages/” folder requested by the \$_REQUEST[‘page’] GET attribute or a sub application request from the \$_REQUEST[‘app’] GET attribute (see Figure 8).

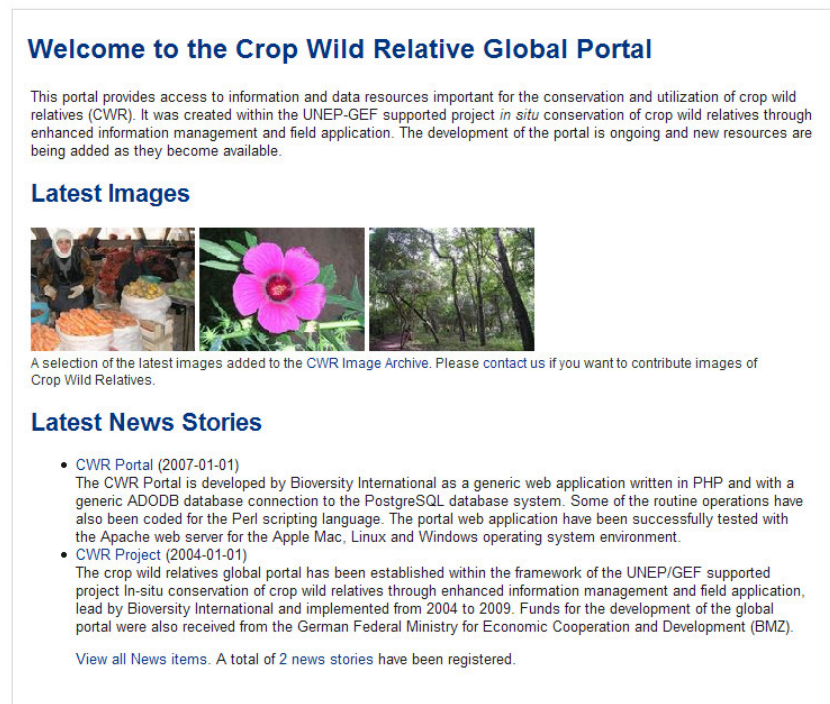
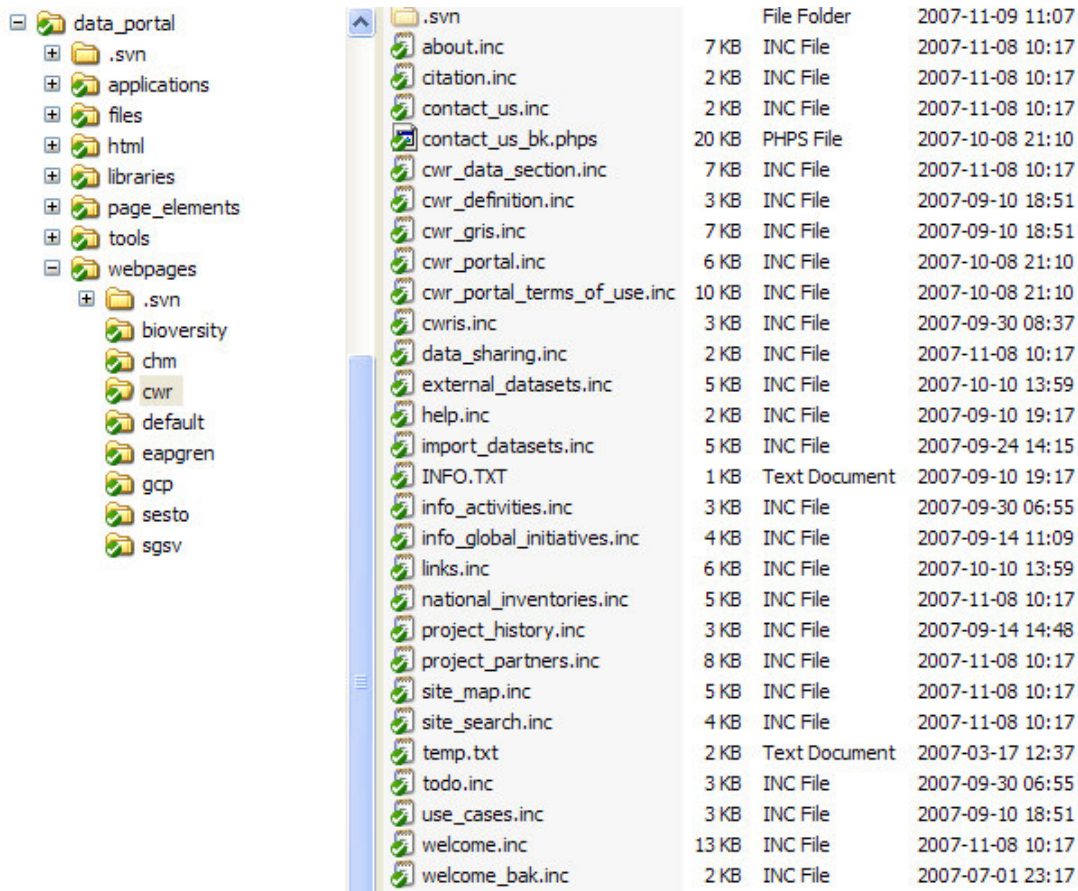


Figure 8, the page middle content frame wraps the data portal content from a sub-application or from a information web page.

Information pages

If the data portal is called from a URL with a GET attribute “page”, `$_REQUEST['page']` the data portal will look for a information web page (file) located in the “data_portal/webpages/<scope>” folder with file extension “.inc” or “.phps” (Figure 9). If such a file is not located, the “data_portal/webpages/” directory will be searched to include the default generic information page. If no file is located an error message will explain this to the user. Add the information web pages you wish to use in your portal implementation to the “data_portal/webpages/<scope>” directory and link to them using the internal link: “LINK TEXT”.



File Folder	2007-11-09 11:07
about.inc	7 KB INC File 2007-11-08 10:17
citation.inc	2 KB INC File 2007-11-08 10:17
contact_us.inc	2 KB INC File 2007-11-08 10:17
contact_us_bk.phps	20 KB PHPS File 2007-10-08 21:10
cwr_data_section.inc	7 KB INC File 2007-11-08 10:17
cwr_definition.inc	3 KB INC File 2007-09-10 18:51
cwr_gris.inc	7 KB INC File 2007-09-10 18:51
cwr_portal.inc	6 KB INC File 2007-10-08 21:10
cwr_portal_terms_of_use.inc	10 KB INC File 2007-10-08 21:10
cwrinc.inc	3 KB INC File 2007-09-30 08:37
data_sharing.inc	2 KB INC File 2007-11-08 10:17
external_datasets.inc	5 KB INC File 2007-10-10 13:59
help.inc	2 KB INC File 2007-09-10 19:17
import_datasets.inc	5 KB INC File 2007-09-24 14:15
INFO.TXT	1 KB Text Document 2007-09-10 19:17
info_activities.inc	3 KB INC File 2007-09-30 06:55
info_global_initiatives.inc	4 KB INC File 2007-09-14 11:09
links.inc	6 KB INC File 2007-10-10 13:59
national_inventories.inc	5 KB INC File 2007-11-08 10:17
project_history.inc	3 KB INC File 2007-09-14 14:48
project_partners.inc	8 KB INC File 2007-11-08 10:17
site_map.inc	5 KB INC File 2007-11-08 10:17
site_search.inc	4 KB INC File 2007-11-08 10:17
temp.txt	2 KB Text Document 2007-03-17 12:37
todo.inc	3 KB INC File 2007-09-30 06:55
use_cases.inc	3 KB INC File 2007-09-10 18:51
welcome.inc	13 KB INC File 2007-11-08 10:17
welcome_bak.inc	2 KB INC File 2007-07-01 23:17

Figure 9, information web pages are loaded from the “data_portal/webpages/<scope>” directory, requested by the `$_REQUEST['page']` GET attribute.

Sub applications

The data portal comes with a set of sub applications you may chose to use for your portal implementation. Add a link to the sub applications you wish to use with the internal link: “LINK TEXT” from one of the navigation menus. You may of course add your own sub applications to the “data_portal/applications/” directory (Figure 10) and call them the same way with the `$_REQUEST['app']` GET attribute value equal to the sub application folder name. Please do not include any <html>, <head>

or <body> HTML tags. You should also remember to create the “main.php” script (“data_portal/applications/<sub application name>/main.php”) to start the new sub application.

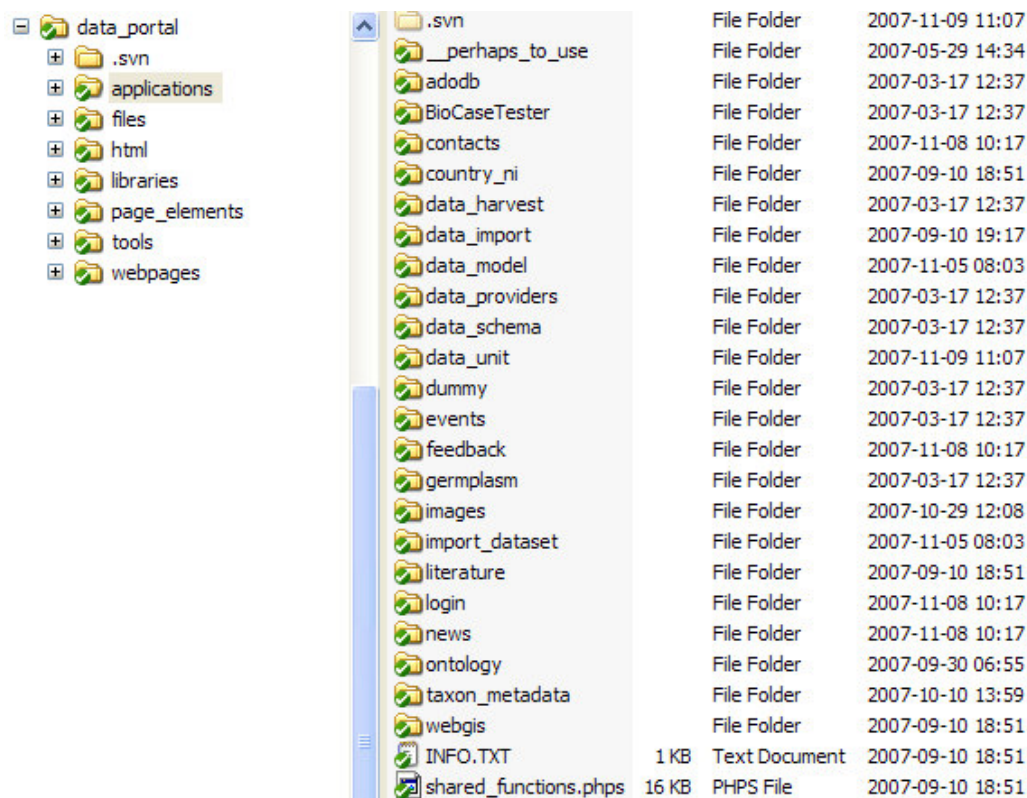


Figure 10, portal content sub applications are loaded from the “data_portal/applications/” directory, requested by the \$_REQUEST[‘app’] GET attribute.

See the user manual for more information of how to use the individual sub applications.

Data harvest routines and methods

The data portal is designed to publish and integrate distributed datasets – not to maintain original source datasets. The target distributed source datasets can be provided as a (set of) simple file(s) or as a more mature web service. The data portal was originally designed to access, scan and index XML data output from the GBIF type BioCASE PyWrapper database wrapper web service. Many relevant and important datasets on genetic resources and related biodiversity data types are not yet available as XML data from web services. Many datasets are still either provided as an online file or as a file provided in by personal contact or as uploaded to the portal web site. The portal application was thus extended for easier import and indexing of also files of a defined data model and following one of the supported file formats.

- Web services
- Simple files

Datasets provided as a XML web service (BioCASE)

The original data portal as derived from the GCP Central Registry application, the Germplasm Clearing House Mechanism (CHM, <http://chm.grinfo.net>). The CHM portal was developed during January to March 2006 (Figure 11).

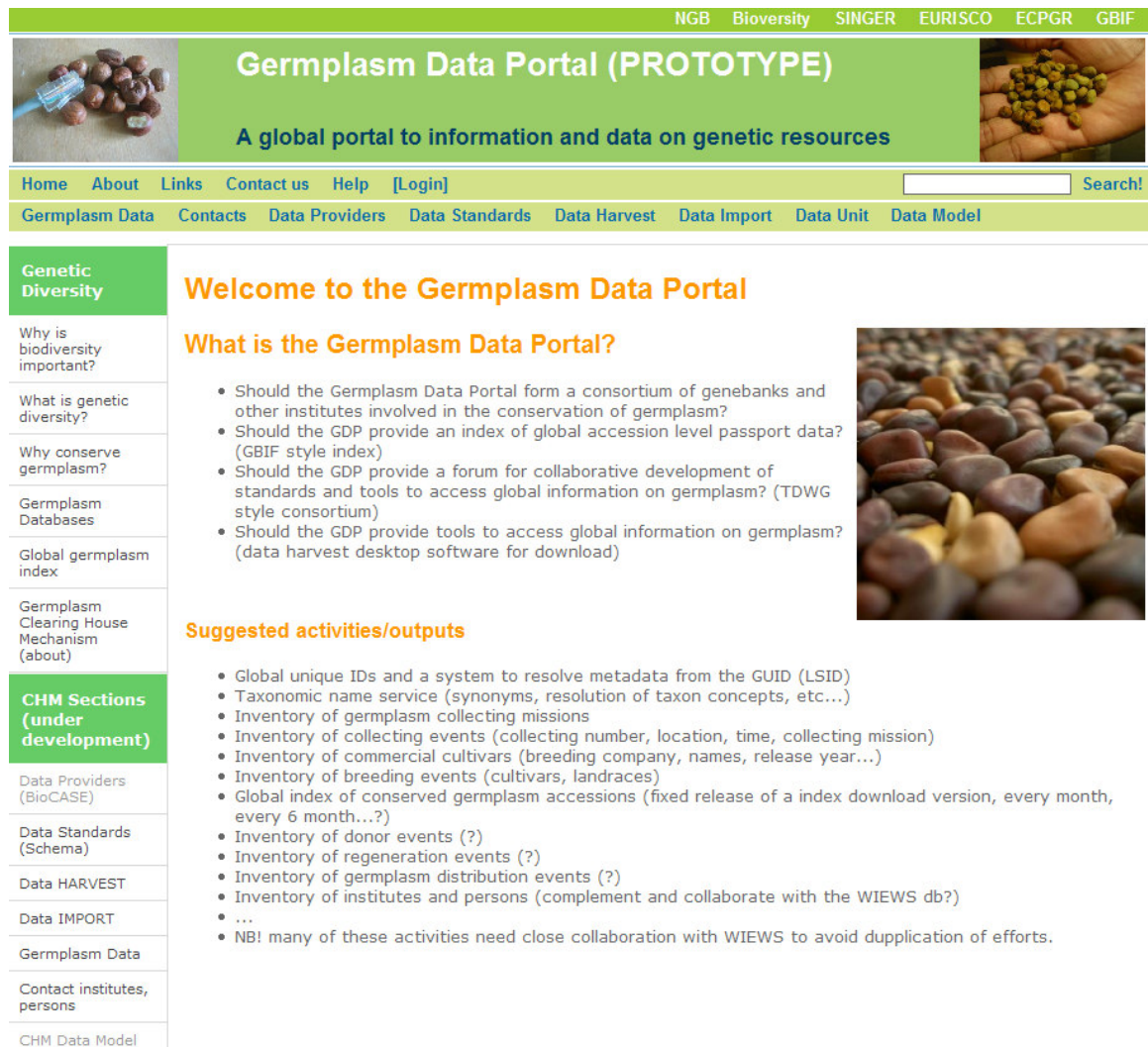


Figure 11, the first version of the data portal was the Germplasm Clearing House Mechanism, designed to access, scan and index XML data from BioCASE database wrapper web service end points.

The steps of the CHM portal are implemented as sub applications and available to other implementations of the data portal (like the CWR, SGSV, EAPGREN etc.). The steps of indexing remote and distributed BioCASE end points includes (1, Figure 12) a list of the

discovery URLs of the data provider services; (2, Figure 13) a list of the supported global data standards used by these data providers to publish the relevant datasets including a mapping of these standards to the data model of the CHM index; (3, Figure 14) methods to formulate the data request query as specified by the BioCASE protocol (request.xml) and harvest the XML data from the selected BioCASE provider service; (4, Figure 15) methods to preview the harvested XML data and extract data values to be imported to the CHM database index (Figure 16).

Data providers

BioCASE data provider entry points (URL)

DSA BioCASE URL	Data Source	Data provider	
AVRDC [N/A]	AVRDC	AVRDC	[Details] QueryForm
Accessions [http://ww3.bgbm.org/biocase/]	IPK, Helmut Knüpfner	BGBM	[Details] QueryForm
CGN [http://www.plant.dlo.nl/biocase/]	CGN, Theo van Hintum	CGN	[Details] QueryForm
ciatcass [http://gene3.ciat.cgiar.org/biocase/]	CIAT, Fernando Rojas	CIAT	[Details] QueryForm
CIP [http://216.244.151.138/biocase/]	CIP, William Roca	CIP	[Details] QueryForm
IITA_BAMBARANUT [http://genebank.iita.org/biocase/]	IITA, Visvanathan Mahalaksmi	IITA	[Details] QueryForm
IITA_CASSAVA [http://genebank.iita.org/biocase/]	IITA, Visvanathan Mahalaksmi	IITA	[Details] QueryForm
IITA_COWPEA [http://genebank.iita.org/biocase/]	IITA, Visvanathan Mahalaksmi	IITA	[Details] QueryForm
IITA_SOYBEAN [http://genebank.iita.org/biocase/]	IITA, Visvanathan Mahalaksmi	IITA	[Details] QueryForm
IITA_WILD_VIGNA [http://genebank.iita.org/biocase/]	IITA, Visvanathan Mahalaksmi	IITA	[Details] QueryForm
IITA_YAM [http://genebank.iita.org/biocase/]	IITA, Visvanathan Mahalaksmi	IITA	[Details] QueryForm
SINGER [http://biocase.grinfo.net/]	SINGER, Samy Gaiji	IPGRI	[Details] QueryForm
MGIS [http://biocase.inibap.org/]	IPGRI-INIBAP, Nicolas Roux	IPGRI-INIBAP	[Details] QueryForm
ICIS [http://www.iris.irri.org:8080/biocase/]	IRRI, Ruairaidh Sackville Hamilton	IRRI	[Details] QueryForm
Iva [http://geifir.ngb.se/biocase/]	LUBI, Isaak Rashal	NGB	[Details] QueryForm
NGB [http://geifir.ngb.se/biocase/]	NGB, Dag Endresen	NGB	[Details] QueryForm
grin [http://198.77.175.232:8080/biocase/]	USDA-GRIN, Quinn Sinnott	USDA-GRIN	[Details] QueryForm

Figure 12, step 1 of the CHM is a list of data provider BioCASE service end points. All the BioCASE DSA URLs are registered to provide the starting point for a data harvest session. A normal UDDI with a standard WSDL style discovery would be a useful extension of this step 1.

Data schema

List [data schema] [schema elements/concepts]

Data schema (xsd)

Schema title	Version	Schema full name	Published	
ABCD	1.20	[ABCD 1.20]	2003 March	[Details] [List concepts]
ABCD	2.06	[ABCD 2.06]	2005 September	[Details] [List concepts]
Darwin Core 2	1.2	[Darwin Core 2 v1.2]	2003-06-13	[Details] [List concepts]
GCP Passport	1.02	[GCP Passport 1.02]	2005-09-15	[Details] [List concepts]
GCP Passport	1.03	[GCP Passport 1.03]	2005-11-30	[Details] [List concepts]
GCP Passport	1.04	[GCP Passport 1.04]	2006-03-08	[Details] [List concepts]
Germplasm CHM	1.0	Germplasm_CHM_1.0	2006-03-13	[Details] [List concepts]
Multi Crop Passport XML		Multi Crop Passport	2005 July	[Details] [List concepts]

Figure 13, step 2 is the list of supported global data standards including their mapping to the implemented CHM data model of the CHM database index.

Data harvest

BioCASE web service entry point

* Data Provider:

[More information about the selected data provider \[here\]...](#)

Data Schema to harvest

* Data schema:

[Mapped schema from the selected provider](#)

[Review/refresh the list of supported schema live \[here\]...](#)

[More information about the selected data schema \[here\]...](#)

Data Harvest Filter condition

Filter concept:

[Mapped concepts from the selected provider](#)

Filter value:

[Use '*' as wildcard](#)

Additional settings

Number of records:

[Limit the number of records to be harvested](#)

Records per page:

[Maximum records per page \(debug\)](#)

[An asterix \(*\) marks the required fields.](#)

[Please login for permissions to start data harvest...](#)

Figure 14, step 3 is the interface to formulate the data request (request.xml) according to the BioCASE protocol. The data harvest methods are developed as a PHP library and can be started either directly from the web interface or from the UNIX prompt command line (or the crontab). The data harvest includes paging of the XML data response from the harvested BioCASE end point when there are more records available than the requested number of records per page (or the maximum allowed records per page the remote BioCASE DSA is configured to allow).

Data import

BioCASE web service entry point

Data provider:

Preview of the harvested records (this page)

Records: Total 500 records harvested.

Switch data import on, after review of the data

Data import:

Harvest settings for previous data harvest instance was

- * BioCASE URL [<http://192.168.181.162/pywrapper.cgi?dsa=SINGER>]
- * Schema [http://www.ipgri.org/schemas/gcp_pass/1.02]
- * Max number of records per page 100
- * Max number pages

Harvest results for previous data harvest instance was

- * Last data harvest 2006-03-21 16:43:06
- * Harvest duration was 9.519 seconds
- * Total records **500**

Local id	Scientific name	Holding institute	Collection name	Longitude	Latitude	Provider
CIATBEAN-G1	Beta vulgaris L.	COL003	CIAT - Bean collection			IPGRI
CIATBEAN-G10003	Beta vulgaris L.	COL003	CIAT - Bean collection			IPGRI
CIATBEAN-G10071	Beta vulgaris L.	COL003	CIAT - Bean collection			IPGRI
CIATBEAN-G10072	Beta vulgaris L.	COL003	CIAT - Bean collection			IPGRI
CIATBEAN-G10073	Beta vulgaris L.	COL003	CIAT - Bean collection			IPGRI
CIATBEAN-G10074	Beta vulgaris L.	COL003	CIAT - Bean collection			IPGRI
CIATBEAN-G10075	Beta vulgaris L.	COL003	CIAT - Bean collection			IPGRI
CIATBEAN-G10077	Beta vulgaris L.	COL003	CIAT - Bean collection			IPGRI
CIATBEAN-G10078	Beta vulgaris L.	COL003	CIAT - Bean collection			IPGRI
CIATBEAN-G10079	Beta vulgaris L.	COL003	CIAT - Bean collection			IPGRI

Preview of 10 records from the last harvested data.

Figure 15, step 4 is the preview of the harvested XML data, extracting selected data values and the import of these values to the CHM database index.

Harvested and Indexed Germplasm Data

[\[default search form\]](#) [\[simple search form\]](#)

Search form

Your search criteria was:

No search criteria given! All files in the registry displayed...

Add search criteria

Add filter

Add columns to the list

Display

10

hits per page.

Go!

Total hits 8212, record 1 to 10 displayed [Next page](#)

Taxon	Accession number	Holding institute	Details
Beta vulgaris L.	CIATBEAN-G1	COL003	[Details]
Beta vulgaris L.	CIATBEAN-G10	COL003	[Details]
Beta vulgaris L.	CIATBEAN-G100	COL003	[Details]
Brassica rapa ssp. oleifera	NGB13106	Nordic Gene Bank	[Details]
Lactuca sativa	NGB4102	Nordic Gene Bank	[Details]
Poa pratensis	NGB2439	Nordic Gene Bank	[Details]
Lolium perenne	NGB13332	Nordic Gene Bank	[Details]
Pisum sativum ssp. sativum	NGB12196	Nordic Gene Bank	[Details]
Lolium multiflorum	NGB15422	Nordic Gene Bank	[Details]
Brassica oleracea var. capitata f. alba	NGB13555	Nordic Gene Bank	[Details]

[Export search results to Excel](#)

Click the column header to sort the list ↓ click same column head again for reverse order ↑

Figure 16, the CHM portal also comes with a search interface to the CHM database index.

Dataset(s) provided as a REST XML web service (GBIF)

The Global Biodiversity Information Facility (GBIF) support the implementation of tools to establish a distributed global network of biodiversity information resources based on the standards developed and maintained by TDWG (Biodiversity Information Standards). The GBIF data portal [<http://data.gbif.org>] harvest data records from this network of distributed biodiversity data providers and index a selected number of descriptors (including the scientific name, geospatial origin, record identifier/catalog number, holding institute etc.). The GBIF data index can be explored from the excellent data search portal, but more important to the germplasm data portal is the web service interface of the GBIF data index [<http://data.gbif.org/tutorial/services>].



Biodiversity
Information
Standards
TDWG

Figure 17, the Global Biodiversity Information Facility (GBIF) maintains a data portal of global distributed datasets on biodiversity based on the standards developed and maintained by TDWG (Biodiversity Information Standards).

The GBIF data portal provides a public web service interface to the harvested and indexed distributed datasets. The GBIF data portal web service interface supports SOAP and REST type interaction. REST (Representational State Transfer) is an architectural style which in practice means that the online web resource is called from a standard URL where each of the parts of the URL divided by the slash (“/”) represent one state. Each of these URL-“parts” can roughly be compared to the XML markup tags of a SOAP XML service request. A REST style service basically means that each unique URL is a representation of some object. And that you can get the contents of that object using an HTTP GET.

The GBIF data portal offer REST web service interfaces for taxon, occurrence records, occurrence density, dataset metadata, data provider metadata and data network metadata level data. An example of the occurrence record REST service request style:

`http://data.gbif.org/ws/rest/occurrence/<action>?<parameter_list>`

An example of the service request style asking for all occurrence records of the species *Allium porrum*:

`http://data.gbif.org/ws/rest/occurrence/count?scientificname=Allium+porrum`



GBIF occurrence web service response

Request details	
service	occurrence
scientificname	Allium porrum
request	count
Number of records matched	1467

-
This document contains data shared through the GBIF Network - see <http://data.gbif.org/> for more information.

All usage of these data must be in accordance with the GBIF Data Use Agreement - see <http://www.gbif.org/DataProviders/Agreements/DUA>

-

For help with this web service, see: <http://data.gbif.org/ws/rest/occurrence/help>

Figure 18, example of GBIF response format:

[<http://data.gbif.org/ws/rest/occurrence/count?scientificname=Allium+porrum>]

An example of the service request style asking for all occurrence records of the species *Allium porrum* with geospatial origin attributes reported (geo-referenced records only):

<http://data.gbif.org/ws/rest/occurrence/count?scientificname=Allium+porrum&georeferenceonly=true&stylesheet=>

```

- <gbif:gbifResponse xsi:schemaLocation="http://portal.gbif.org/ws/response/gbif
  http://data.gbif.org/ws/rest/occurrence/schema http://purl.org/dc/elements/1.1/
  http://data.gbif.org/schema/dc.xsd http://purl.org/dc/terms/ http://data.gbif.org/schema/dcterms.xsd
  http://www.w3.org/1999/02/22-rdf-syntax-ns# http://data.gbif.org/schema/rdf.xsd
  http://www.w3.org/2002/07/owl# http://data.gbif.org/schema/owl.xsd
  http://rs.tdwg.org/ontology/voc/Common# http://data.gbif.org/schema/tcom.xsd
  http://rs.tdwg.org/ontology/voc/TaxonOccurrence# http://data.gbif.org/schema/TaxonOccurrence.xsd
  http://rs.tdwg.org/ontology/voc/TaxonConcept# http://data.gbif.org/schema/TaxonConcept.xsd
  http://rs.tdwg.org/ontology/voc/TaxonName# http://data.gbif.org/schema/TaxonName.xsd">
- <gbif:header>
  - <gbif:statements>
    - This document contains data shared through the GBIF Network - see http://data.gbif.org/
      for more information. All usage of these data must be in accordance with the GBIF Data Use
      Agreement - see http://www.gbif.org/DataProviders/Agreements/DUA -
    </gbif:statements>
    <gbif:help>http://data.gbif.org/ws/rest/occurrence/help</gbif:help>
    <gbif:request>count</gbif:request>
    <gbif:parameter name="service" value="occurrence"/>
    <gbif:parameter name="coordinatestatus" value="true"/>
    <gbif:parameter name="scientificname" value="Allium porrum"/>
    <gbif:parameter name="request" value="count"/>
    <gbif:parameter name="stylesheet" value=""/>
    <gbif:summary totalMatched="90"/>
  </gbif:header>
</gbif:gbifResponse>

```

Figure 19, example of GBIF response format:
<http://data.gbif.org/ws/rest/occurrence/count?scientificname=Allium+porrum&georeferencedonly=true&stylesheet=>

It is the XML mark-up tag attribute “totalMatched” of the mark-up tag named “gbif:summary” we are interested in for the germplasm data portal. In the example for *Allium porrum*, only geo-referenced records, we find this attribute to report 90 such occurrence records indexed by the GBIF data portal. Figure 20 show the PHP source code implementation of the germplasm data portal to access and extract the total count of occurrence records for a given species and/or a country of origin (with the filter condition for geo-referenced records only on and off). The germplasm data portal administrator may invoke the refreshing of the GBIF occurrence count from the web interface (Figure 21). The function can also be called from the command line or added to the crontab (UNIX-like systems) for a scheduled automatic refresh of the taxon level summery number of GBIF occurrence records (Figure 22).

```

function gbif_count_occurrences ($taxon_name, $country_iso2, $georeferenced_only = 'false') {
/** function: gbif_count_occurrences
 * Connect to the GBIF Data Portal and calculate summary species level or country level record count
 * INPUT: Species name, Country ISO-2 Code, Georeferenced_only (true/false)
 */
$totalMatched = ""; // init variable
if ($country_iso2) : $country_iso2 = strtoupper($country_iso2); endif;
$rest_url = "http://data.gbif.org/ws/rest/occurrence/count?";
$rest_url .= "stylesheet="; // switch off stylesheet browser formatting
if ($taxon_name) : $rest_url .= "&scientificname=" . urlencode($taxon_name); endif;
if ($country_iso2) : $rest_url .= "&originisocountrycode=" . strtoupper($country_iso2); endif;
$rest_url .= "&georeferencedonly=" . $georeferenced_only; // switch for georeferenced ONLY on/off
$file_content = file_get_contents($rest_url); // Read the GBIF Data Portal web service response as REST URL
$xml = new SimpleXMLElement($file_content);
$xml->registerXPathNamespace('gbif', 'http://portal.gbif.org/ws/response/gbif'); // Register namespace
$result = $xml->xpath('//gbif:summary[1]');
foreach ($result as $data_node) {
    foreach ($data_node->attributes() as $key => $value) {
        if ($key == 'totalMatched') {
            $totalMatched = (string) $value;
        } // end if key totalMatched
    } // end foreach data_node attribute
} // end foreach result data_node
# echo "<xmp>\n" . $file_content . "</xmp><hr />\n"; // DEBUG
return $totalMatched;
}

```

Figure 20, this is the PHP code to access the GBIF data portal REST web service interface.

Total hits 1846, record 1 to 20 displayed [Next page](#)

ID	Taxon	GBIF records	Georeferenced	Previous update
133	<i>Abelmoschus angulosus</i>	2 -> 2	0	2007-10-11 16:26:27
134	<i>Abelmoschus ficulneus</i>	55 -> 55	3 -> 3	2007-10-11 16:26:27
135	<i>Abelmoschus moschatus</i>	433 -> 433	134 -> 134	2007-10-11 16:26:28
15	<i>Abutilon</i>	388 -> 388	132 -> 132	2007-10-11 16:26:29
137	<i>Abutilon avicennae</i>	514 -> 514	1 -> 1	2007-10-11 16:26:29
136	<i>Abutilon theophrasti</i>	1 994 -> 1 994	1 616 -> 1 616	2007-10-11 16:26:30
138	<i>Acalypha amentacea</i>	215 -> 215	30 -> 30	2007-10-11 16:26:31
139	<i>Acalypha ciliata</i>	54 -> 54	10 -> 10	2007-10-11 16:26:32
140	<i>Acalypha fruticosa</i>	79 -> 79	15 -> 15	2007-10-11 16:26:32
141	<i>Acalypha hispida</i>	162 -> 162	25 -> 25	2007-10-11 16:26:33
142	<i>Acalypha lanceolata</i>	114 -> 114	22 -> 22	2007-10-11 16:26:33
143	<i>Acalypha racemosa</i>	41 -> 41	18 -> 18	2007-10-11 16:26:34
144	<i>Acalypha supera</i>	2 -> 2	0	2007-10-11 16:26:34
146	<i>Acanthophyllum gypsophylloides</i>	0	0	2007-10-11 16:26:34
145	<i>Acanthophyllum paniculatum</i>	0	0	2007-10-11 16:26:35
147	<i>Acanthophyllum tadshikistanicum</i>	0	0	2007-10-11 16:26:35
148	<i>Achyranthes aspera</i>	788 -> 788	432 -> 432	2007-10-11 16:26:36
149	<i>Achyranthes bidentata</i>	169 -> 169	45 -> 45	2007-10-11 16:26:37
150	<i>Achyranthes diandra</i>	0	0	2007-10-11 16:26:37
151	<i>Adenanthra bicolor</i>	0	0	2007-10-11 16:26:37

Figure 21, the function in the previous figure (Figure 20) to refresh the cached summary number of species occurrences from the GBIF web service can be invoked from the germplasm data portal web interface.

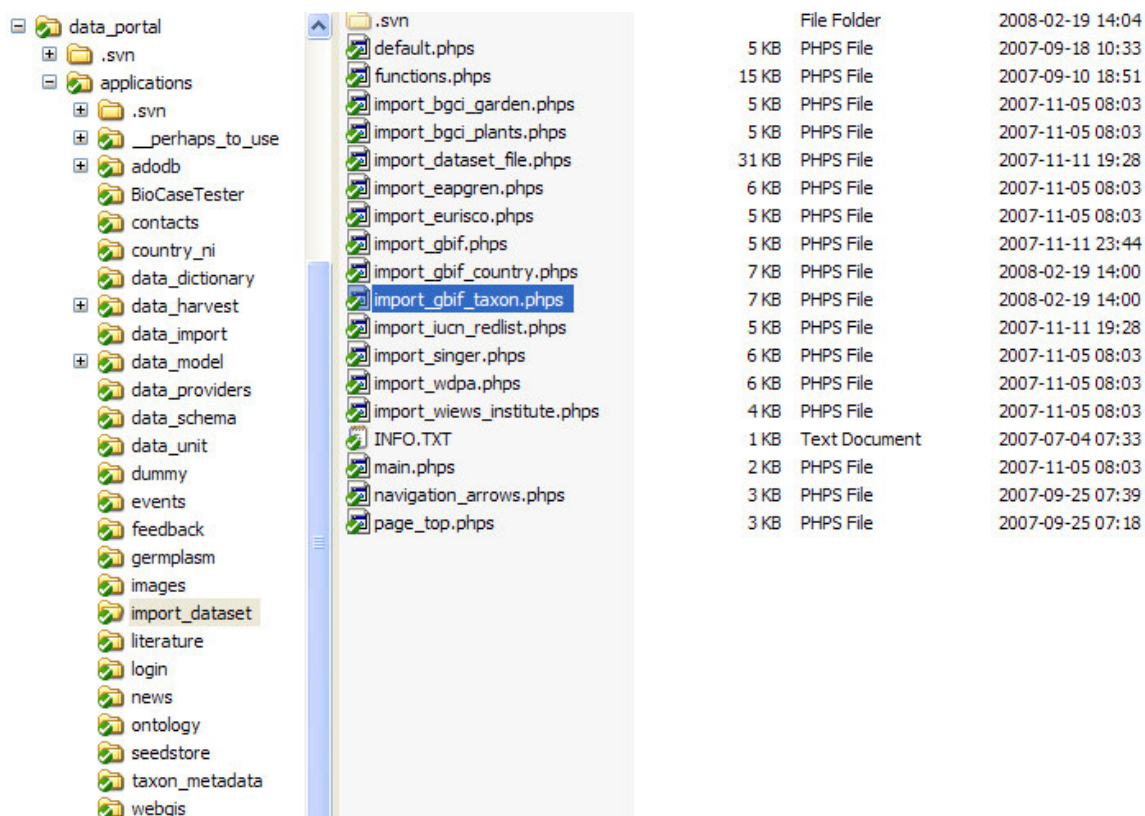


Figure 22, the function (Figure 20) to refresh the count of occurrence records for a species from the GBIF REST web service can be invoked from a PHP script “data_portal/applications/import_datasets/import_gbif_taxon.php”. This script can be executed from the command line or added to the crontab for a scheduled automatic refresh (... may require some minor update of the current version of the script).

Update of the summary metadata on the taxon and country unit level may also be updated for an individual species or country from the URL

[./index.php?app=import_dataset&inc=import_gbif&data_unit=taxon&taxon_name={species_name}] and

[./index.php?app=import_dataset&inc=import_gbif&data_unit=country&country_name={country_name}]. The link is displayed for logged in users from the corresponding species and country metadata page.

Datasets provided as a simple file

Many relevant and interesting datasets of importance to genetic resources management are still only available as simple files. We are still a long way from seeing a wider implementation of standard web services for even half of the relevant source datasets. Many of these datasets are maintained and updated using local database systems and local data models and then regularly exported and provided as a simple file. In the best cases this regularly updated simple file is published from a stable URL available either from the HTTP or the FTP protocol. In other cases the simple file needs to be extracted

from a manual user interaction with an online information system, often “protected” by a data disclaimer or a data use license the user will need to (manually) accept. And yet another data exchange alternative is when the simple file is provided by personal interaction (e.g. email attachment) or manually uploaded to the data portal web site with a web form file upload or a FTP file upload.

The simple file may also be provided as a variety of file formats. The XML data format is not very common for simple files. In the best cases the simple file is provided as tab delimited data values with line breaks between individual data records. The comma separated values (CSV) format is also common. The benefit of this format is that quote marks can be used to wrap more complex data values, which again can also be a source of error when mistakes (including un-escaped quotes inside the wrapped data value and missing closing quote marks) are transmitted. The proprietary spreadsheet format of the Microsoft Excel files is also popular; as well as even simply sharing a complete Microsoft Access database file. The dBase file format is yet another common file format, which at least have a published open file format protocol.

Another challenge with interpretation and extract of data from datasets provided as simple files is that these are often provided as compressed files. Most common is the ZIP format (file.zip) and the GZ and BZ2 format (file.gz, file.tar.gz, file.tar.bz2). GZ compressed files is often a compression of a TAR ball, used to combine several files and/or directories in one file.

The methods and routines of the data portal to access interpret and index datasets provided as a simple file attempts to meet all these challenges mentioned above.

Import of external datasets

- if dataset is online => wget online dataset source file
- if dataset file is compressed => un-compress and save file to the portal (zip, gz and tar formats are supported)
- convert the (un-compressed) dataset file to tab-separated text
 - if the dataset file is of the comma separated values, CSV spreadsheet format => convert to tab-separated text
 - if the dataset file is of the Excel spreadsheet format => convert to tab-separated text
 - if the dataset file is of the dBase database table format => convert to tab-separated text
- convert the tab-separated text version of the dataset to a SQL INSERT script
- import the SQL INSERT version of the downloaded dataset to the portal database
- some datasets are also post-processed after import to the database
- calculation of taxon and country level summary metadata

An overview of the steps to access, download, convert and import an external reference dataset to the germplasm data portal is also visualized in Figure 24.

As input configuration to the “import dataset” sub-application of the data portal you will need to give a short acronym for your dataset to be used as the default folder name and file name below. If the source dataset is published by the dataset provider from a online URL, you need to describe this to the configuration file. If your source dataset file is not available from a stable URL, you may need to download the file and save it to the correct folder manually. You will need to give the character encoding of the source dataset file. If no character encoding is given the data portal will attempt to guess the encoding. If you source dataset file is compressed as a .zip or a .gz (.gz, .tar.gz, tar.bz2), the data portal will sense this (using the file extension) and un-compress the file. Other compression formats are not (yet) supported and you will need to un-compress the file manually. You may want to give the file type. If the un-compressed source dataset file has the “correct” file extension, you may leave the file type o be decided by the extension. The data portal supports tab delimited files (.tab, .txt), MS Excel spreadsheet (.xls), and comma separated values (.csv). XML data and dBase files are partially supported but this will be developed further. Other file types will give an error message and you will need to convert them to tab delimited text manually. The list of supported file types will be extended. You will need to give the directory path to the folder on your server where you want the (temporary) files to be placed during the import dataset routines. The directory “data_portal/files/data_import/<dataset_acronym>” will be used as the default. NB! Make sure the web server have write permission to this folder if you wish to use this routine!

```
dataset_acronym: institute_views
source_url: http://apps3.fao.org/wiews/export.zip
source_file_name: export.zip
import_path: /usr/local/data_portal/files/data_import/institute_views/
dataset_file_name: export.txt
dataset_file_type: csv
dataset_encoding: utf8
text_file_name: institute_views.txt
sql_file_name: institute_views.sql
import_table: institute_views
```

Figure 23, the configuration attributes for the “import dataset” sub-applications, showing the attributes for the WIEWS Institute as example.

The “import dataset” configuration file is located in the directory of the sub-application itself: “data_portal/applications/import_dataset/import_<dataset_acronym>.phps”. The web interface for the dataset import is invoked by the GET attribute “inc=import_<dataset_acronym>” e.g.:
“http://servername.net/index.php?app=import_dataset&inc=import_views_institute”.

You may execute ALL the “import_dataset” steps from one single link or execute individual steps from the data portal web interface. All steps can be completed manually following the instructions below or added to an automatic server routine from the

crontab. The sub-routines for each step are implemented with the Perl scripting language. Some supporting external software need to be installed on your server as described below.

Step 1, download dataset file from online source URL.

This step will download the source dataset (using the attribute “source_url” from the dataset configuration file). You may of course download the source dataset file manually and save the file to the directory “data_portal/files/data_import/<dataset_acronym>” of your data portal installation directory. The data portal web interface will provide you with information of the last time the source file was downloaded (file date), as well as the file size. E.g.: Source dataset file [export.zip] was last modified on **November 01 2007 12:22:15** (0.6493 MByte). The data portal web interface will also describe the manual download link and the full path to the directory where you are expected to save the file.

Step 2, un-compress the source dataset file, if needed (zip, tar, gz, bz2).

This step will un-compress the source dataset file if the file have the file extension “.zip”, “.gz”, “.tar.gz” or “.tar.bz2”. Other compression formats like the “.rar” is not (yet) supported. You may of course un-compress the source dataset file manually and save the un-compressed file to “data_portal/files/data_import/<dataset_acronym>”.

Step 3, Convert the (un-compressed) dataset file to tab-separated text

The routine to convert the dataset to the appropriate SQL INSERT script expects tab-separated text input. The data portal comes with support for automatic conversion of MS Excel spreadsheet (.xls), and comma separated values (.csv) to tab-separated text (“data_portal/tools/bin/xls2txt.pl”, “data_portal/tools/bin/csv2txt.pl”). XML data and dBase files are partially supported but this will be developed further. Support for the conversion of additional file formats can be added as needed. You may of course manually convert the dataset file to tab-separated text and save to the “data_portal/files/data_import/<data_acronym>” directory.

Step 4, recode the tab-text dataset file to Unicode, if needed

The recommended implementation of the data portal is for the Unicode (utf-8) encoding. You may of course define your database also with other encoding schemas. The data portal ill by default transform the source dataset file to the utf-8 character encoding. The Perl script “data_portal/tools/bin/text_recode.pl” is used for this operation ad will require the external application “recode” (<http://www.gnu.org/software/recode/>) to be installed at the server. You may perhaps prefer to use the iconv API (<http://en.wikipedia.org/wiki/Iconv>) for the recoding of the dataset character encoding. You may of course recode the tab-delimited text file manually (for example using iconv)

and save the recoded file under the same file name
("data_portal/files/data_import/<data_acronym>/<data_acronym>.txt").

Step 5, transform the tab-delimited dataset file to SQL INSERT script

The data portal may automatically transform tab-delimited text data values to a SQL INSERT script. Records are expected to be separated by a line-break and data values by the tab character. The first row of the file is expected to hold the column/field names. The Perl script "data_portal/tools/bin/text2psql.pl" is used for this operation. If you want to create the SQL INSERT script manually, please save the file as "data_portal/files/data_import/<data_acronym>/<data_acronym>.sql".

Step 6, IMPORT dataset to the database

This step will import the data from the SQL INSERT script to the database. The Perl script "data_portal/tools/bin/sql2db.pl" is used for this operation. This script will start by deleting (DROP) of the previous table with the <dataset_acronym> name before the table is recreated and loaded with the new data values. The "sql2db.pl" script is developed for the PostgreSQL database system only. You may of course perform this step manually using for example the command: "psql -d<database> -f <SQL INSERT file name>".

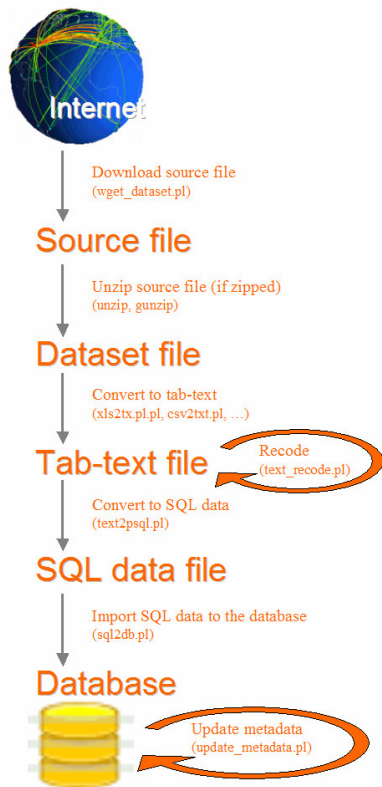


Figure 24, a summary flow of the steps to access, download, convert and import an external reference dataset to the germplasm data portal.

Import of external reference datasets (examples from the CWR Global Portal)

The data portal as implemented for the CWR Global Portal is prepared for (automatic) import of a number of external reference datasets. A few details for each of these datasets will be described here as examples. Other implementations of the germplasm data portal application than the CWR Global Portal may use the very same configuration to import these external reference datasets if this is useful.

WIEWS Institute

The WIEWS institute table holds the “Institute code” used as the standard identifier for institutes with activities relevant to the Genetic Resources community. The dataset is maintained by FAO (Food and Agriculture Organization of the United Nations). The dataset is available from the stable online URL: <http://apps3.fao.org/wiews/export.zip>, and have been so for years. The WIEWS institute dataset is provided as utf-8 and zip compressed. The un-compressed dataset file is “export.txt” and provided as comma separated values where all the data values are wrapped in double quote marks. The WIEWS institute dataset file use simple carriage returns (CR, \r, 0x0D, Mac OS 9 style line break) to separate the records. The carriage return characters are converted to line feed (LF, \n, 0x0a, UNIX style line break) with the Perl script “data_portal/tools/bin/text_line_break.pl”. At the time of writing this technical manual the WIEWS institute dataset includes a CR+LF (Windows style line break) inside of the data value for “URL” in the record for INSTCODE “CHE082” (The Swiss Agency for Development and Cooperation). Due to problems to parse this record it will unfortunately be excluded during the dataset import. An exception for this record is hard coded in the “csv2txt.pl” Perl script. The WIEWS Institute dataset will successfully be imported from a fully automatic procedure.

BGCI Garden and BGCI Plants

The dataset from the Botanical Gardens Conservation International (BGCI) is not (yet) made available from a stable online URL. The dataset imported to the CWR Global portal will be shared through personal communication with BGCI and made available as an email attachment. The provided source dataset files are “bgci_garden.csv”, “bgci_plants.csv” and “bgci_plant_to_garden.csv”. The BGCI dataset is provided as semi-colon separated values, partly wrapped in double quote marks. CR+LR are used to separate records (Windows style line breaks). For many of the records the values from multiple columns directly after the “plantid” seems to be wrapped inside the same double quote marks. For this reason the quote marks are ignored as defining the wrapping of data values for the bgci_plants dataset. A significant number of records do not contain data and also ignored. Exceptions for data records difficult to parse are hard coded in

“csv2txt.pl” and in text2psql.pl”. The BGCI dataset as last provided will import successfully if new dataset files are saved as “data_portal/files/data_import/bgci/bgci_garden.csv” and “data_portal/files/data_import/bgci/bgci_plants.csv”. The “import data” sub application includes an (automatic) routine to update the taxon level and country level CWR metadata. For this purpose the “data_portal/tools/bin/update_metadata.pl” is used. You may of course execute this Perl script manually (or add to you crontab) as “update_metadata.pl bgci_garden” and “update_metadata.pl bgci_plants”.

IUCN Red List

The IUCN Red List dataset is not (yet) online from a stable URL. The dataset imported to the CWR Global portal was manually extracted from the IUCN web portal (<http://www.iucnredlist.org>) and downloaded as “comma separated values where only the data values for scientific name are wrapped in double quote marks. Line feed characters (n, LF, 0x0A, UNIX style line breaks) are used to separate data records. The “import data” sub application (automatic) routine to update the taxon level and country level CWR metadata uses the “data_portal/tools/bin/update_metadata.pl” Perl script. To execute this Perl script manually (or add to you crontab) use the command “update_metadata.pl iucn_redlist”. Note that the data model of the exported IUCN red list dataset may change in later version, so attention is advised.

WDPA, World Database on Protected Areas

The World Database on Protected Areas (WDPA) is not (yet) available as a standard data file from a stable URL. The dataset imported to the CWR Global Portal was manually extracted from the ESRI shape files of protected areas provided online from the UNEP-WCMC, WDPA web site. The WDPA dataset (ESRI shapefiles) require the user to manually accept the data license during download and can thus not be automated. From the WDPA dataset in ESRI shapefile format the descriptive text data on the protected areas was extracted in dBase format and converted to tab delimited text manually. The process to extract data from the dBase file format can be done using an external application like e.g. dbf [<http://berg-systeme.de/dbf.html>], [<http://pkgsr.se/wip/dbf>] or the dbf2psql [<ftp://ftp.ngb.se/pub/linux/db/>] application. Using MS Excel may cause problems with the 65 536 record limit. You are recommended to use the ESRI shapefiles for set 2 and set 4 (only point data and without polygon data) as these contains a unique list of all the protected areas. To update this dataset save a refreshed tab delimited list of protected areas as “data_portal/files/data_import/unesp_wdpa/wdpa_protected_area.txt” and execute step 5 and 6 (text2psql.pl and sql2db.pl). Note that you will need to make a new data use license with UNEP-WCMC before you do this! Note also that the current data use license for the WDPA dataset in the CWR Global Portal is time limited!

EURISCO

The EURISCO dataset is maintained by Bioversity International on behalf of ECPGR. The EURISCO database is available as a BioCASE web service and shared with GBIF. The current metadata for the EURISCO dataset was manually extracted directly from (a copy of) the original EURISCO database as maintained at Bioversity. The Perl script “update_metadata.pl eurisco” may be used to update the CWR taxon level and country level metadata. The recommended extension of indexing the EURISCO dataset would be to use the BioCASE or the TAPIR/PyWrapper3 web service interface.

SINGER

The SINGER dataset is maintained by Bioversity International on behalf of the CGIAR, SINGER. The SINGER database is available as a BioCASE web service and shared with GBIF. The current metadata for the SINGER dataset was manually extracted directly from (a copy of) the original SINGER database as maintained at Bioversity. The Perl script “update_metadata.pl singer” may be used to update the CWR taxon level and country level metadata. The recommended extension of indexing the SINGER dataset would be to use the BioCASE or the TAPIR/PyWrapper3 web service interface.

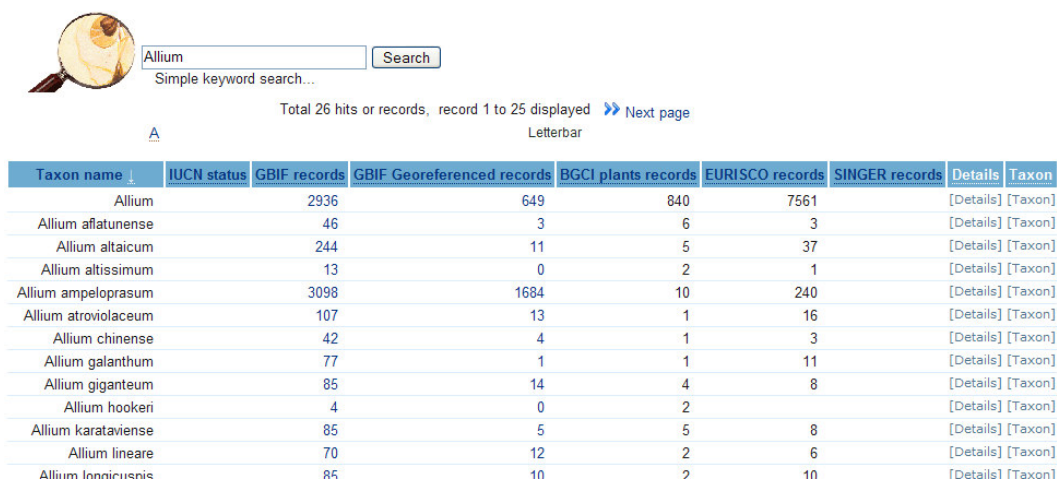
Taxon and country unit level summary metadata

The external datasets for the CWR Global Portal is summarized from individual taxon and a country unit level search interface (see Figure 25) with taxon (see Figure 26) and country pages.

Taxon metadata

Search taxon metadata by keyword

You will find more search options with the [\[advanced search form\]](#).



The search interface shows a magnifying glass icon, a search bar with the text "Allium", and a "Search" button. Below the search bar, it says "Simple keyword search...". To the right, it indicates "Total 26 hits or records, record 1 to 25 displayed" and a "Next page" link. Below this, there is a "Letterbar" with the letter "A". The main part of the interface is a table with the following data:

Taxon name	IUCN status	GBIF records	GBIF Georeferenced records	BGCI plants records	EURISCO records	SINGER records	Details	Taxon
Allium		2936	649	840	7561		[Details]	[Taxon]
Allium aflatunense		46	3	6	3		[Details]	[Taxon]
Allium altaicum		244	11	5	37		[Details]	[Taxon]
Allium altissimum		13	0	2	1		[Details]	[Taxon]
Allium ampeloprasum		3098	1684	10	240		[Details]	[Taxon]
Allium atroviolaceum		107	13	1	16		[Details]	[Taxon]
Allium chinense		42	4	1	3		[Details]	[Taxon]
Allium galanthum		77	1	1	11		[Details]	[Taxon]
Allium giganteum		85	14	4	8		[Details]	[Taxon]
Allium hookeri		4	0	2			[Details]	[Taxon]
Allium karataviense		85	5	5	8		[Details]	[Taxon]
Allium lineare		70	12	2	6		[Details]	[Taxon]
Allium innanicum		85	10	2	10		[Details]	[Taxon]

Figure 25, search interface (simple keyword search) for taxon level metadata from the indexed external datasets on CWR resources.

Allium schoenoprasum

Taxon summary data	
Family:	Alliaceae
Full scientific name:	Allium schoenoprasum L.
Genus:	Allium
Species:	schoenoprasum
Taxonomical reference:	1753, Sp. Pl., :301
 The 2006 IUCN Red List of Threatened Species. Accessed on 2007-07-09. [Data use guidelines]	
IUCN status:	Not found in the IUCN Red List (NE, Not Evaluated)
 Number of records from GBIF [http://data.gbif.org] Accessed on 2007-07-20. [Data use guidelines]	
GBIF Records:	3 670 records
Georeferenced:	2 762 records
 Number of records from BGCI [http://www.bgci.org] BGCI accessed on 2007-07-06. [Data use guidelines]	
BGCI Records:	14 records
Countries:	TODO
 Number of records from SINGER [http://singer.grinfo.net] SINGER was accessed on 2007-07-23. [Terms of use]	
SINGER Records:	Not found in the SINGER dataset
Georeferenced:	No georeferences records available from the SINGER data index
Countries:	TODO
 Number of records from EURISCO [http://eurisco.ecpgr.org] EURISCO was accessed on 2007-07-23. [Terms of use]	
EURISCO Records:	172 records
Georeferenced:	68 records
Countries:	TODO
Search information provided from FAO portals	
 [Search FAOLEX] for legal legislation on Allium schoenoprasum Data use agreement	
 [Search EcoPort] for more information on Allium schoenoprasum Data use agreement	
 [Search ECOLEX] for more information on Allium schoenoprasum Data use agreement	
 [Search AGRIS/CARIS] for more information on Allium schoenoprasum	
Search information from other sources	
 [Search Google] for Allium schoenoprasum [Search Google Images] for Allium schoenoprasum	
 [Search Species2000] for Allium schoenoprasum Species2000 is part of Catalogue of Life.	
 [Search Wikispecies] for Allium schoenoprasum Wikispecies is an open, free, wiki directory of species.	
 [Search ILDIS] for more information about Allium schoenoprasum International Legume Database & Information Service	
 [Search TROPICOS] (Missouri Botanical Garden) for more information about Allium schoenoprasum	
 [Search flickr] for pictures of Allium schoenoprasum	
 [Search picsearch] for pictures of Allium schoenoprasum	
[Edit Taxon Metadata] [Update GBIF taxon metadata] (These links to update the metadata is displayed ONLY for logged in users...)	

Figure 26, example of a taxon level metadata detail page for *Allium schoenoprasum*.

Country metadata

Search form

Your search criteria was:

Country name contains

Add search criteria:

contains

Add columns to the list Display hits per page.






Total 19 hits or records, record 1 to 19 displayed

[B](#) [E](#) [F](#) [H](#) [I](#) [K](#) [L](#) [M](#) [P](#) [S](#) [U](#) Letterbar

Country name	Country ISO-3	Continent name	GBIF records	BGCI plants records	EURISCO records	SINGER records	Details	Country
British Indian Ocean Territory	IOT	Asia	6413			2	[Details] [Country]	
British Virgin Islands	VGB	North America	6930	6		57	[Details] [Country]	
Eritrea	ERI	Africa	1939		13	38	[Details] [Country]	
France, Metropolitan	FXF						[Details] [Country]	
Haiti	HTI	North America	21501		3	236	[Details] [Country]	
Holy See (Vatican City)	VAT	Europe	6604		2		[Details] [Country]	
Italy	ITA	Europe	330506	4377	24905	2809	[Details] [Country]	
Kuwait	KWT	Asia	1094		3		[Details] [Country]	
Lithuania	LTU	Europe	7358	2750	2487	15	[Details] [Country]	
Mauritania	MRT	Africa	1105		24	110	[Details] [Country]	

Figure 27, search interface (advanced search) for country level metadata from the indexed external datasets on CWR resources.

Country summary data

Italy	
Continent name: Europe	
Country ISO-3 Code: ITA	
Country ISO-2 Code: IT	
	Number of total records from GBIF [http://data.gbif.org] Last accessed on 2008-01-30. [Data use guidelines]
GBIF Records: 330 506 records (not limited to CWR taxa)	
	Number of plant records from BGCI [http://www.bgci.org] BGCI accessed on 2007-07-06. [Data use guidelines]
BGCI plants: 4 377 plant records (not limited to CWR taxa)	
BGCI gardens: 104	
	Number of records from SINGER [http://singer.grinfor.net] Accessed on 2007-07-23. [Terms of use]
SINGER: 2 809 accessions (not limited to CWR taxa)	
Georeferenced: 525 accessions in SINGER are georeferenced	
	Number of ex situ accessions from EURISCO [http://eurisco.ecpgr.org] Accessed on 2007-07-23. [Terms of use]
EURISCO: 24 905 accessions (not limited to CWR taxa)	
Georeferenced: 1 454 accessions in EURISCO are georeferenced	
	Number of protected areas in Italy from the World Database of Protected Areas (WDPA). The WDPA is a joint venture of UNEP and the IUCN, produced by UNEP-WCMC and the IUCN WCPA working with governments and collaborating NGOs. It is updated continuously providing the most current data on protected areas worldwide. The following link gives you access to this database: [http://www.unep-wcmc.org/wdpa/] The WDPA dataset was last accessed by the CWR global portal on 2007-05-21. [Data use guidelines]
WDPA: 1 042 protected areas in Italy (reported in WDPA of UNEP WCMC)	



Italy map (from the CIA World Factbook)

Search information from FAO

 **WIEWS** Number of institutes with reported PGR activity in Italy from WIEWS
[\[http://apps3.fao.org/wiews/wiews.jsp\]](http://apps3.fao.org/wiews/wiews.jsp)
 Accessed on 2007-03-15. [\[Data use guidelines\]](#)

WIEWS: 73 institutes in Italy have PGR activity (reported in WIEWS)

 **FAOLEX** [\[Search FAOLEX\]](#) for legal legislation on **Italy**
[Data use agreement](#)

 **EcoPort** [\[Search EcoPort\]](#) for more information on **Italy**
[Data use agreement](#)

 **ECOLEX** [\[Search ECOLEX\]](#) for more information
[Data use agreement](#)

 **AGRIS/CARIS** [\[Search AGRIS/CARIS\]](#) for more information on **Italy**

[[Edit Country Metadata](#)] [[Update GBIF country metadata](#)] (These links to update the metadata is displayed **ONLY** for logged in users.)

Manual update of taxon and country unit level metadata

```

cwr=# \d country_metadata
          Column          |          Type          |          Table "public.country_metadata"          |          Modifiers          |
-----+-----+-----+-----+-----+
country_metadata_id | integer | | not null default nextval('country_metadata_country_met |
adata_id_seq'::regclass) | | | | |
country_name | character varying(255) | | | | |
country_iso3 | character(3) | | | | |
country_iso2 | character(2) | | | | |
continent_name | character varying(255) | | | | |
gbif_records | integer | | | | |
gbif_occurrences | integer | | | | |
gbif_dtm | timestamp(0) without time zone | default now() | | | |
bgci_records | integer | | | | |
bgci_gardens | integer | | | | |
bgci_dtm | timestamp(0) without time zone | default now() | | | |
eurisco_records | integer | | | | |
eurisco_occurrences | integer | | | | |
eurisco_dtm | timestamp(0) without time zone | default now() | | | |
singer_records | integer | | | | |
singer_occurrences | integer | | | | |
singer_dtm | timestamp(0) without time zone | default now() | | | |
ni_records | integer | | | | |
ni_occurrences | integer | | | | |
ni_dtm | timestamp(0) without time zone | default now() | | | |
wdpa_areas | integer | | | | |
wdpa_dtm | timestamp(0) without time zone | default now() | | | |
country_id | integer | | | | |
country_image_map | character varying(255) | | | | |
country_image_flag | character varying(255) | | | | |
creusr | character varying(32) | default "current_user"() | | | |
credtm | timestamp(0) without time zone | default now() | | | |
updusr | character varying(32) | default "current_user"() | | | |
upddtm | timestamp(0) without time zone | default now() | | | |
Indexes:
    "country_metadata_pkey" PRIMARY KEY, btree (country_metadata_id)
    "country_metadata_country_metadata_id_idx" UNIQUE, btree (country_metadata_id)

cwr=# UPDATE country_metadata set bgci_records = 4377 WHERE country_name = 'Italy';

```

The CWR Global Portal also has an edit interface for such manual update of taxon and country level metadata.

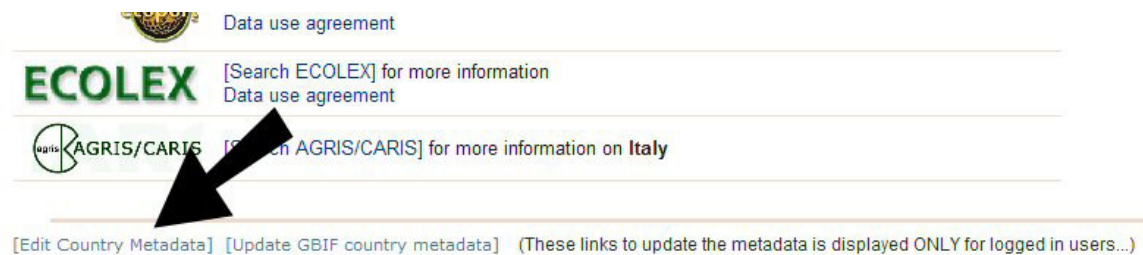


Figure 30, here is the link to the “edit country metadata” form. This link is ONLY displayed for logged in users.

Country level metadata

Search the CWR Country metadata by keyword [GO!](#)

Or use the [\[Advanced search form\]](#)

Update Country Metadata for Italy

GBIF dataset	
GBIF Records:	<input type="text" value="330506"/>
GBIF Georeferenced Records:	<input type="text"/>
GBIF last accessed:	<input type="text" value="2008-01-30 22:10:07"/>
BGCI dataset	
BGCI Records:	<input type="text" value="4377"/>
BGCI gardens:	<input type="text" value="104"/>
BGCI last accessed:	<input type="text" value="2007-07-06 00:00:00"/>
EURISCO dataset	
EURISCO Records:	<input type="text" value="24905"/>
EURISCO Georeferenced Records:	<input type="text" value="1454"/>
EURISCO last accessed:	<input type="text" value="2007-07-23 00:00:00"/>
SINGER dataset	
SINGER Records:	<input type="text" value="2809"/>
SINGER Georeferenced Records:	<input type="text" value="525"/>
SINGER last accessed:	<input type="text" value="2007-07-23 00:00:00"/>
NI dataset	
NI Records:	<input type="text"/>
NI Georeferenced Records:	<input type="text"/>
NI last accessed:	<input type="text"/>
WPDA dataset	
WPDA Areas:	<input type="text" value="1042"/>
WPDA last accessed:	<input type="text" value="2007-05-21 00:00:00"/>


An asterisk (*) marks the required fields.

[Register country metadata](#)

Figure 31, this is the edit form for country level metadata. You would normally update these data points from the (semi-) automatic update routines for external datasets. For example the GBIF summary metadata is very easy to update (per unit as well as for more units) from the link located directly next to the link to this form from the taxon and country level metadata detail pages...

Data dictionary

Data displayed in the data portal may have a more descriptive column names or data labels than the basic database table column name, if defined in the data dictionary (see Figure 32 and Figure 33). The logged in users will find a link to the data dictionary from the left menu (or from the site map) to define data dictionary descriptions (see Figure 34 and Figure 35).

Taxon name 	IUCN status	GBIF records	GBIF Georeferenced records	BGCI pl
Abelmoschus angulosus		0	0	
Abelmoschus ficulneus		55	3	
Abelmoschus moschatus		437	193	
Abutilon		469	172	
Abutilon avicennae		750	1	

Number of records indexed by the GBIF data portal

Figure 32, example of descriptive column names and mouse over column tip as defined from the data dictionary for a data unit list view.

About

CWR Data

Taxon level metadata

Country level metadata

Contact institutes, persons

Image Archive

News stories

Literature about CWR resources

Web Mapping, visualization

CWR Ontology

[Import Dataset]

[Data dictionary]

[Explore the Data Model]

[Browse CWR Data Units]

Taxon metadata

Taxon metadata details

Taxon name: *Allium schoenoprasum*

IUCN ID:

IUCN status:

GBIF TaxonConceptKey:

GBIF records: 3670

GBIF Georeferenced records: 2762

GBIF data last updated: 2008-02-12 17:42:30

BGCI ID:

BGCI plants records: 14

Bgci dtm: 2008-02-12 18:04:33

EURISCO ID:

EURISCO records: 172

EURISCO Georeferenced records: 68

SINGER ID:

SINGER records: 0

SINGER Georeferenced records: 0

CWR NI Records:

NI Georeferenced records:

TaxonID PRIMARY KEY: 208

Taxon GUID:

Full scientific name: *Allium schoenoprasum* L.

Genus: *Allium*

Species: *schoenoprasum*

Subtaxon:

Rank: species

Hybrid:

Remarks:

Figure 33, example of descriptive column names from the data dictionary for a data unit detail view.

About

CWR Data

Taxon level metadata

Country level metadata

Contact institutes, persons

Image Archive

News stories

Literature about CWR resources

Web Mapping, visualization

CWR Ontology

Import Dataset

Data dictionary

Explore the Data Model

Browse CWR Data Units

Data dictionary

The data dictionary provides description and definitions of database tables and columns as well as coded data values.

- db_tables describes the tables in the database
- db_columns describes the column fields or attributes of the tables
- db_values describes coded column values including their decoding
- data_objects is an abstractions of db_tables (db_tables are data_objects)
- data_descriptors is an abstractions of the db_columns (db_columns are descriptors)
- data_model describes the raw data structure including tables and columns - this is the previous primitive data dictionary

[\[tables\]](#)
[\[descriptors, columns\]](#)
[\[coded values\]](#)

Table details

Table name: taxon_metadata

Descriptive name: Taxon level metadata

Description, caption: Taxon (species) unit level summary metadata from the external reference datasets on CWR resources

Columns: 32

Records: 1846

[Update table description]

[View Columns]

[BACK to the list of tables]

Database table fields, descriptors (for taxon_metadata)

Column name	Descriptive column name		Data Model
bgci_dtm	Column NOT described!	[Not described] [Edit]	NOT in DM [Details DM] [Edit DM]
bgci_id	Column NOT described!	[Not described] [Edit]	BGCI ID [Details DM] [Edit DM]
bgci_records	Column NOT described!	[Not described] [Edit]	BGCI plants records [Details DM] [Edit DM]
credtm	Column NOT described!	[Not described] [Edit]	Not Described [Details DM] [Edit DM]
creusr	Column NOT described!	[Not described] [Edit]	Not Described [Details DM] [Edit DM]
eurisco_id	Column NOT described!	[Not described] [Edit]	EURISCO ID [Details DM] [Edit DM]
eurisco_occurrences	Column NOT described!	[Not described] [Edit]	EURISCO Georeferenced records [Details DM] [Edit DM]
eurisco_records	Column NOT described!	[Not described] [Edit]	EURISCO records [Details DM] [Edit DM]
full_scientific_name	Column NOT described!	[Not described] [Edit]	Not Described [Details DM] [Edit DM]
gbif_dtm	Column NOT described!	[Not described] [Edit]	GBIF data last updated [Details DM] [Edit DM]
gbif_occurrences	Column NOT described!	[Not described] [Edit]	GBIF Georeferenced records [Details DM] [Edit DM]
gbif_records	Column NOT described!	[Not described] [Edit]	GBIF records [Details DM] [Edit DM]

Figure 34, start the data dictionary description by a description of the database table (step 1).

Update column description (for taxon_metadata.bgci_dtm)

Column description

Descriptive column name: BGCI data last updated

Description of column: Timestamp when the metadata value from the BGCI external reference dataset was last updated

Data unit, class:

Remarks

Remarks:

An asterix (*) marks the required fields.

Register description

[Return to the table detail page]

Figure 35, next describe the individual columns using the [Edit DM] links from the table description detail page. You may also consider updating the column description from the [Edit] link as well. Work is in progress for a new improved data dictionary model based on this concept.

Frequently asked questions:

** May I install my own local implementation of the germplasm data portal?*

* The germplasm data portal is open source, free to use for any purpose and GPL2 licensed. You are thus most welcome to implement the complete data portal application or take parts of it. You are free to distribute the application and/or the source code further. But if you wish to distribute a modified version, please contact the Nordic Gene bank or Bioversity International.

** Will the data portal work with Tomcat?*

* The data portal is not a Java application and will not work with Apache Tomcat. If you wish to serve both PHP applications and Java applications from your online web server, you will need to install both Apache Httpd and Apache Tomcat (or analogue web application servers). You may serve Java applications through the Apache Httpd with a connector/plugin for the Apache Tomcat server. I do not believe it is as easy to serve PHP applications from a similar approach through the Apache Tomcat web server.

** Does the germplasm portal follow the W3C guidelines?*

* The data portal attempts to follow the [W3C Web Accessibility Guidelines \(WCAG10\)](http://www.w3c.org/TR/2000/NOTE-WCAG10-TECHS-20001106/) [http://www.w3c.org/TR/2000/NOTE-WCAG10-TECHS-20001106/]. The data portal attempts to implement a separation of content and layout using the CSS, cascading style sheet definitions for generation of the layout.

Software used by or useful to the data portal

- Apache web server httpd server [<http://httpd.apache.org>]
- PHP: Hypertext Preprocessor script programming language [<http://www.php.net>]
- PostgreSQL database server [<http://www.postgresql.org>]
- ADOdb Database Abstraction Library for PHP [<http://adodb.sourceforge.net>]
- Perl programming language [<http://www.perl.org/>]
- Subversion version control system Code Repository alternative to CVS [<http://subversion.tigris.org>]
- Recode character set conversion library. The open source recode library is used for import of non-UNICODE external datasets. [<http://directory.fsf.org/recode.html>]

- ICONV character set conversion library. The open source iconv library is an alternative to recode used for import of non-UNICODE external datasets.
[<http://www.gnu.org/software/libiconv/documentation/libiconv/iconv.1.html>]
- GNU Wget [<http://www.gnu.org/software/wget/>]
- ImageMagick convert. The command line tool convert from the ImageMagick is used by the Simple Image Archive.
[<http://www.imagemagick.org/script/convert.php>]

References:

- CWR Global Portal, [<http://cwrint.grinfo.net>]
- CWR Global Portal User Manual
[http://cwrint.grinfo.net/files/cwr/CWR_Portal_Search_Manual.pdf]
- SESTO genebank information system [<http://www.nordgen.org/sesto/>]
- GCP Central Repository [<http://gcpcr.grinfo.net>]
- Germplasm Clearing House Mechanism (CHM) [<http://chm.grinfo.net>]
- REST web service style
[http://en.wikipedia.org/wiki/Representational_State_Transfer]
- W3C Web Accessibility Guidelines (WCAG10)
[<http://www.w3c.org/TR/2000/NOTE-WCAG10-TECHS-20001106/>]
- SGSV Portal, Svalbard Global Seed Vault data portal
[<http://www.nordgen.org/sgsv/>]